



Kyran D. Mish, Center Director

The work presented here represents a transition period in computational engineering at Lawrence Livermore National Laboratory (LLNL). This year's effort marks the emergence of an integrated administrative structure designed to recognize the common aspects of mechanical and electrical engineering as well as their fundamental differences. This administrative organization is the Center for Computational Engineering, integrating the former mechanical and electrical engineering thrust areas into a more coherent unit, with a clear focus on the development and use of engineering simulation software as an enabling technology for LLNL.

Our FY-99 research portfolio still reflects the natural divisions present in the engineering directorate, though some of the projects demonstrate the cross-cutting multidisciplinary nature of computational engineering. A fine example of such an effort is the droplet-based net-form manufacturing work, which couples mechanical and thermal response into a simulation code capable of guiding experimentation into the emerging field of droplet-based manufacturing techniques. Three other examples of multiple-use technology are the development of finite-element preprocessors for the EIGER electromagnetics suite, the investigation into data-management software for computational mechanics, and the research into scalable solver technology for the NIKE3D mechanics code.

Each of these research efforts is oriented toward one branch of engineering, but has substantial potential benefits to other engineering disciplines at LLNL.

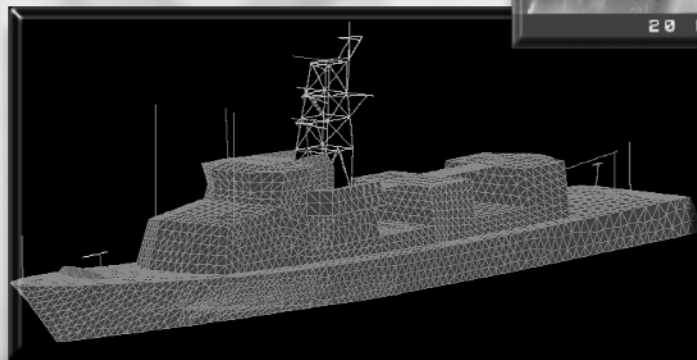
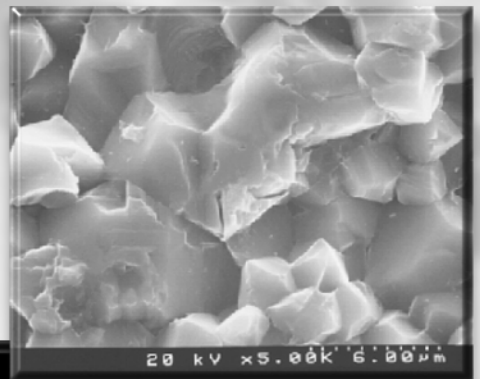
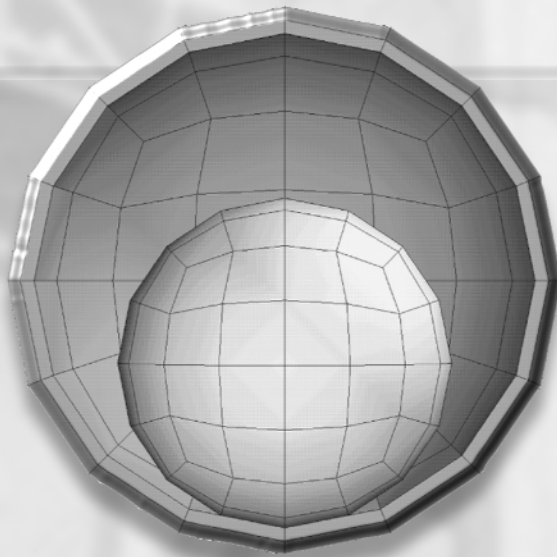
Mechanical engineering research includes fundamental work on material response, such as the reliability analyses of brittle materials and the simulation of anisotropic inelasticity, as well as developmental research on mathematics and computation, such as the improvements in implicit temporal integration algorithms for structural response in NIKE3D, or the slidesurface interface efforts developed for the DYNA3D code.

Electrical engineering enterprises include the development of femtosecond-resolution laser/RF/x-ray pulses for accelerator technology improvements, and software usability efforts developed for the TIGER time-domain electromagnetics code. Finally, the emerging discipline of automata methods in engineering is aptly demonstrated by our pioneering work in Lattice-Boltzmann simulations for chemical engineering and microfluidics.

This nascent computational engineering organization at LLNL will continue to grow in the future, and will both cement the areas of commonality in engineering and reinforce the natural differences. The mission of the center is to build a community of software engineers who are both broad and deep in their understanding of engineering computation.

Center for Computational Engineering

4



4. Center for Computational Engineering

Overview

Kyran D. Mish, Center Director

High-Precision Droplet-Based Net-Form Manufacturing

Arthur B. Shapiro and Melissa Orme 4-1

Improved Implicit Structural Finite Element Algorithms

Michael A. Puso 4-5

Post-Processing and Data Management Software Development for Computational Mechanics

Douglas E. Speck 4-11

Linear Equation Solvers for Parallel Computational Mechanics

Robert M. Ferencz and Chandrika Kamath 4-17

EIGER Preprocessor and User Interface

Robert M. Sharpe, Nathan J. Champagne, and Andrew J. Poggio 4-23

Strength and Reliability of Brittle Materials

Robert A. Riddle, Chol K. Syn, Michael J. O'Brien, Steve Duffy, Joe Palko, and Eric Baker. 4-27

Lattice Boltzmann Modeling of Spherical Solutes in Confined Domains

David S. Clague 4-35

Modeling of Anisotropic Inelastic Behavior

Daniel J. Nikkel, Jr., Deepak S. Nath, Arthur A. Brown, and James Casey 4-39

Development of Laser-Driven Photocathode Injector and Femtosecond-Scale Laser Electron Synchronization for Next Generation Light Sources

Gregory P. Le Sage, Thomas E. Cowan, Todd R. Ditmire, and J. Rosenzweig 4-47

A Segment-to-Segment Sliding Interface Algorithm for DYNA3D

Jerry I. Lin 4-49

Usability Enhancements for TIGER

David J. Steich, Gerald J. Burke, Jeffery S. Kallman, S Terry Brugger, and Daniel A. White. 4-53

Enhanced Fluid Dynamics Capability and Multidisciplinary Coupling in ALE3D

Rose C. McCallen, Timothy A. Dunn, John S. Dolaghan, and Richard G. Couch. 4-63

High-Precision Droplet-Based Net-Form Manufacturing

Arthur B. Shapiro
*New Technologies Engineering
 Mechanical Engineering*

Melissa Orme
*University of California at Irvine
 Irvine, California*

In collaboration with the University of California at Irvine (UCI), we are working on a new technology that relies on the precise deposition of nanoliter molten-metal droplets that are targeted onto a substrate by electrostatic charging and deflection. In this way, 3-D structural materials can be manufactured microlayer by microlayer. Because the volume of the droplets is small, they rapidly solidify on impact, bringing forth a material component with fine grain structures which lead to enhanced material properties such as strength. Lawrence Livermore National Laboratory (LLNL), with its unique expertise in computational modeling and large-scale computer resources, modeled the system to gain insight into manufacturing process control. UCI, with its state-of-the-art experimental facility, provided computer model validation and demonstrated manufacturing feasibility.

Introduction

This new manufacturing process has the potential to fabricate parts with complex geometry, using novel alloys, and possessing improved strength. The very rapid solidification of nanoliter-sized metal droplets leads to finer grains which results in a part possessing higher strength than achievable by conventional casting processes. New and novel alloys can be created by co-jetting droplet streams using metals of different droplet size and deposition rates. Additionally, the microstructure can be varied at different locations in the fabricated part by our ability to control the droplet size, deposition rate, and temperature.

Using multiple nozzles with droplet deposition rates of 40,000 droplets/s would provide a means for rapid prototyping metal parts. Currently, there are several manufacturing techniques using plastic, but no adequate methods exist for rapid prototyping parts using metal. This capability will benefit LLNL's design engineers building one-of-a-kind metal parts or small production quantities.

Progress

Figure 1 is a schematic of the droplet manufacturing facility at UCI. A crucible at the top of the

apparatus contains liquid metal which flows through an orifice forming a capillary stream into the lower vacuum chamber. A method for generating deterministic droplet patterns from capillary stream breakup has been developed by Orme. By applying specific amplitude-modulated disturbances with an

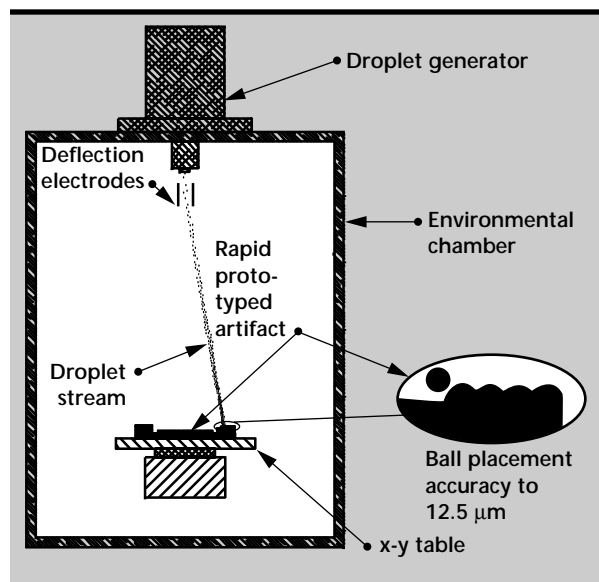


Figure 1. Schematic depicting the High Precision Droplet-Based Net-Shape Manufacturing facility at UCI.

arbitrary modulation to a viscous capillary stream, predictable and flexibly controlled patterns of droplets can be obtained. The disturbance is created by a piezoelectric transducer above the orifice. The metal droplets are charged and their trajectory is controlled by deflection electrodes. The substrate itself can be moved by a controllable x-y positioning table for fabricating planar 3-D parts, or the substrate rotated to fabricate axisymmetric parts as shown in **Fig. 2**.

The parts in **Fig. 2** were fabricated from a former prototype facility using 200- μm solder droplets. The scope of this work is to push the technology so that higher melting point metals such as aluminum can be used and smaller droplet sizes (5 to 50 μm) can be formed to fabricate thin walled parts of interest to LLNL.

During 1998, UCI began moving the technology forward from solder droplets to more interesting materials with higher melting points, such as aluminum. A new facility came on-line in August 1998. By the end of the year, metallurgical investigations of sectioned aluminum-droplet-on-substrate samples from initial shake-down runs had revealed that the grain structure was equiaxed, the grain size ranged from 120 to 150 μm , the average porosity was about 3%, and oxides and other impurities were present. Tensile tests were promising: the measured yield stresses (8900 psi) were greater than both the yield stress for the starting aluminum ingot (6816 psi) and published handbook values for aluminum (6800 psi).

A significant fabrication challenge this year was to overcome the inclusion of oxides and other contaminants in the droplet stream, which would result in defective parts. **Figure 3** shows a droplet formed “stalagmite” (on the right) with a significant presence of oxides indicated by the dark mottled exterior. A filter pack was designed to fit at the bottom of the liquid metal crucible. After four design iterations, we successfully filtered the metal stream and reduced the oxides, as shown by the shiny stalagmite on the left in **Fig. 3**.

During the shake-down runs, we also noted a loss of control on the deposited location of the droplets. This was caused by electrostatic interaction between the closely spaced charged droplets. The solution was to charge alternate droplets in the stream and capture the uncharged droplets following a straight trajectory by a “gutter” to prevent them from striking the substrate. The charged droplets, with a curved trajectory, avoid capture by the gutter. We have perfected the operating procedure for creating a uniform stream of droplets. We are concentrating now on making components. We have made three cylinders from three different runs but haven’t honed in on the correct temperature or speed conditions. We have

some really porous or really “globby” cylinders depending on our conditions. The poor quality of these cylinders is a good illustration of the need for precise operating conditions, and shows the importance of the modeling effort to guide the experiments.

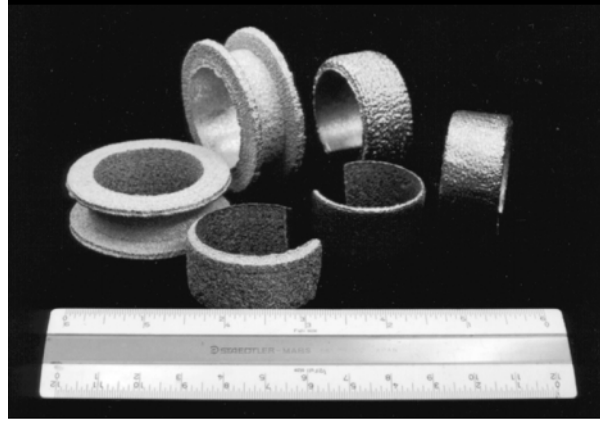


Figure 2. Axisymmetric parts fabricated from a prototype facility using 200- μm solder droplets.



Figure 3. Aluminum stalagmites fabricated by directing a stream of aluminum droplets downward onto a substrate. The stalagmite on the right shows the presence of oxides indicated by the dark mottled exterior. A significant challenge was to design a filter to reduce the oxides and other contaminants. This was successfully accomplished as demonstrated by the shiny stalagmite on the left.

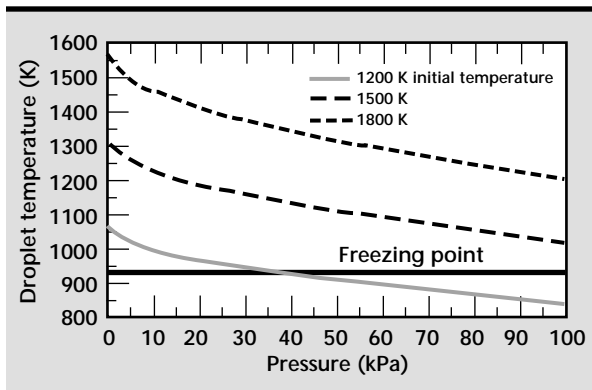
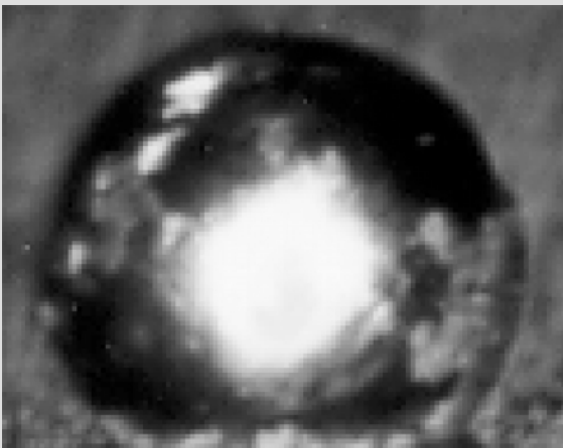


Figure 4. Temperature drop of a 189- μm aluminum droplet with an initial velocity of 9 m/s in nitrogen as a function of chamber pressure. Such curves helped define the process parameters of crucible temperature and chamber pressure that would result in a liquid droplet with a specified superheat when striking the substrate.

(a)



(b)

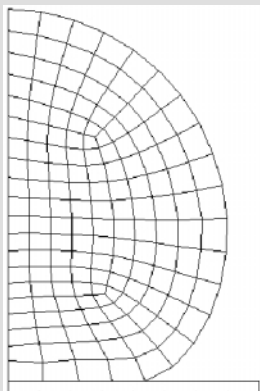


Figure 5. Experimental sessile drop (a) used to determine the material properties for liquid aluminum for our finite element model shown in (b).

During the last year, LLNL pursued process modeling both on the micro-scale of an individual droplet striking a substrate and on the larger system scale to predict velocity and temperature drop during time of flight. We also investigated material constitutive models that would capture the low strength of the metal in the liquid state and transition to the higher strength solid.

Empirical equations were found in the literature to predict the temperature and velocity drop during time of flight. These equations also include the effect of the wake of one droplet on a trailing droplet. Salvador Aceves developed the MELISSA code to solve these equations, which was further developed by Wayne Miller into the code HOTDROP.¹ Figure 4 shows the temperature drop of a 189- μm aluminum droplet with an initial velocity of 9 m/s in nitrogen as a function of chamber pressure. The bottom curve shows that a droplet with an initial temperature of 1200 K will solidify before striking the substrate at chamber pressures above 40 kPa. Such curves helped define the process parameters of crucible temperature and chamber pressure that would result in a liquid droplet with a specified superheat when striking the substrate.

Our next step is to incorporate these empirical equations into our finite element heat transfer code TOPAZ. This will allow us to do system process modeling. Our approach is to “birth” finite elements (but without modeling the impact process—see next paragraph) at the droplet arrival time defined by its velocity and at the arrival temperature. These elements will then lose energy to the environment, previously deposited material, and the substrate. In this way we can predict the cooling rate of the deposited material and gain insight into the microstructure.

On the microscale, we faced a challenge in predicting the shape of a droplet upon impact. The difficulty existed because of our attempt to model a liquid with an elastic-plastic constitutive model. Such a model would be applicable after the droplet solidified. After many failed simulation attempts with ALE3D and DYNA3D to model the impact, we tried to model a stationary liquid drop deforming under gravity (a sessile drop as shown in Fig. 5).

The material constants, such as Young’s modulus, that resulted in the best fit between the analysis and experiment will be used to model the liquid. These properties will be modified for the solid phase according to a temperature dependent function found in the literature for aluminum. We plan to re-investigate the use of ALE3D and DYNA3D in predicting the deformation of a droplet on impact.


Our greatest modeling challenge is to predict the coupled thermal-mechanical response of a droplet impact. We have coupled the solid mechanics code DYNA3D with TOPAZ3D. The hardest part was developing a thermal-mechanical contact algorithm. The algorithm developed has been verified by modeling the heat transfer between two sliding blocks. Our next step is to model droplet impact. Also, other features still need to be implemented, such as, conversion of plastic work to heat.

Validation is critical to our success. The literature is rich in papers investigating the splat of a liquid droplet without solidification.² We found one paper describing an energy balance method that predicted the maximum splat diameter. We had good success in using this equation to predict the results in other papers. This theoretical equation was modified to include solidification³ and predicted a 15% greater spread diameter than measured in an experiment. We will use this equation for validation of our constitutive model under development and results from the coupled DYNA-TOPAZ code effort.

Future Work

We plan to concentrate on manufacturing simple geometric shapes, such as cylinders, and improve the process to decrease defects such as porosity. We will use the system modeling code to determine process parameters.

References

1. W. O. Miller (1999), "HOTDROP Operation Guidelines," Lawrence Livermore National Laboratory, Livermore, California (UCRL-CODE-99040), July.
2. S. M. Aceves, A. B. Shapiro, and V. Sahai (1999), "An accuracy evaluation for the Madejski splat quench solidification model," Lawrence Livermore National Laboratory, Livermore, California (UCRL-JC-133757), March.
3. N. Hadjiconstantinou (1999), "Estimating the maximum splat diameter of a solidifying droplet," Lawrence Livermore National Laboratory, Livermore, California (UCRL-JC-133712), March. 



Improved Implicit Structural Finite Element Algorithms

Michael A. Puso
Defense Technologies Engineering
Mechanical Engineering

We developed new algorithms for contact and implicit dynamics. These algorithms were implemented into the structural mechanics finite element code NIKE3D. The new algorithms provided a more robust and efficient code and improved accuracy. A cursory description of the algorithms along with representative numerical examples are presented here.

Introduction

Implicit nonlinear structural analysis has been used for many years. Nevertheless, it can still be difficult to perform on many problems due to a variety of numerical issues. Implicit finite element analysis has been used most often for determining quasistatic response of structural models, whereas explicit finite element codes have typically been used for nonlinear dynamics problems. Many structural dynamics problems occur over time spans that are quite long for explicit analysis due to the Courant time step limitation, but are too nonlinear or big for implicit dynamic analysis. Seismic analysis, reentry vehicles and transportation container vibration analysis are examples of problems in the "gray area," that is, highly nonlinear, long term dynamics problems.

Progress

Our work solves some of the numerical problems inherent to the implicit method.

The three fundamental difficulties for implicit analysis are as follows:

1. Problem size: Implicit analysis requires the solution to a large system of linear equations which is both memory- and computation-intensive.
2. Nonlinearity: Convergence of the nonlinear solution algorithm is severely impacted by the presence of contact surface unilateral constraints.
3. Stability: For nonlinear problems, Newmark's method is not, in general, unconditionally stable.

New parallel linear solvers have been implemented into NIKE3D and others are in the works for dealing with the large linear systems that are to be handled. Furthermore, a new line of enhanced strain

elements was implemented into NIKE3D this year which provides coarse mesh accuracy. These are *fully* integrated elements based on extensions of our work in *single-point* elements last year.¹ With these elements, an analyst can use a much smaller mesh for the same amount of accuracy as a fine mesh with the older element technology. In this report, we highlight the new developments we have made in the areas of *convergence* and *stability*.

Convergence

The primary culprits for convergence difficulties are the nonlinearities from contact surface interactions due to the non-smooth nature of the contact surfaces. The finite element contact surface comprises bilinear patches or facets. This discretization causes jumps in contact forces at facet edges and vertices making it very difficult for global convergence of the Newton nonlinear solution algorithms.

To alleviate this problem, we have used 3-D bilinear Hermitian surfaces to interpolate the bilinear surface mesh, providing C1 continuity between facets. A 2-D representation of this is shown in **Fig. 1**. This provided a vast improvement in convergence of the contact algorithm.

For example, a solid sphere is contained within a hollow sphere and pressed against the bottom by a vertical pressure (**Fig. 2**). With the new smoothing algorithm we get superior convergence compared to the old method as demonstrated by the residual energy norm produced during the nonlinear iteration loop, as shown in **Table 1** for a given time step. The smoothed representation of the finite element mesh model for the sphere is shown in **Fig. 3**. The smoothed mesh shown in **Fig. 3b** was produced

using Mathematica.² The grid lines are just the level of discretization used for visualization (the true mesh is totally smooth).

In the next example, a single block element is pushed down and then dragged along a surface (**Fig. 4**). This problem particularly illustrates the improved performance when slave nodes encounter new facets during sliding. With the old algorithm, contact forces experience a significant jump when a node changes facets during sliding due to the new normal direction of the facet. The smoothed surface for the configuration in **Fig. 4b** is shown in **Fig. 4c**. The new algorithm provides quadratic convergence as demonstrated by the energy norm results during nonlinear iterations shown in **Table 2**.

Stability

Using Newmark's method for example, the implicit structural dynamics method can be shown to be unconditionally stable for small deformation analysis. For general nonlinear dynamic analysis, the implicit method can become numerically unstable and often does in the presence of large rigid body rotations or severe contact impact. This is a pathological result of the discrete time integration scheme and is not to be confused with a mechanical instability.

With linear analysis, the spectral method is typically used for stability analysis. This is performed by calculating the eigenvalues of the amplitude matrix formed by the recursive integration algorithm. In

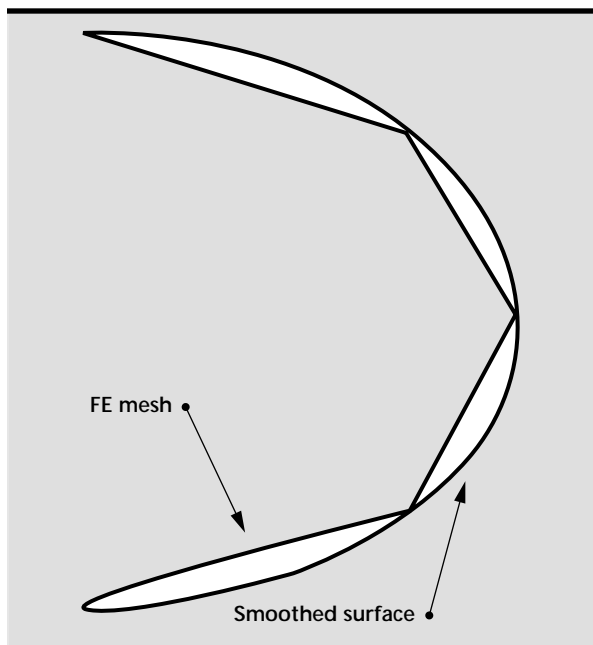


Figure 1. Finite element faceted surface and interpolated smooth surface.

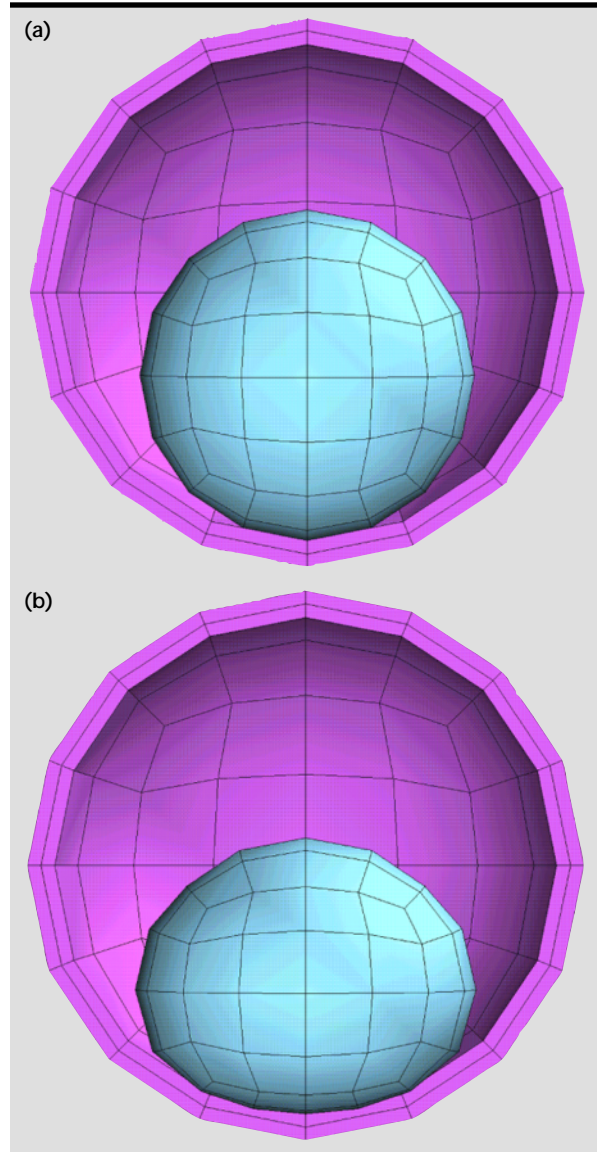


Figure 2. Solid sphere pushed against bottom of hollow sphere.

Table 1. Energy norm results for spheres.

Iteration #	New method	Old method
1	1.3394×10^2	1.3862×10^2
2	1.0054×10^{-1}	1.0287×10^2
3	2.4924×10^{-11}	8.7395×10^{-2}
4	4.9348×10^{-17}	9.2101×10^{-7}
5		9.0672×10^{-7}
6		2.9843×10^{-7}
7		6.0875×10^{-7}
8		5.2122×10^{-7}
9		6.1894×10^{-7}
10		1.1556×10^{-6} (diverged)

nonlinear implicit analysis, an amplitude matrix is not available, so another method is necessary. In particular, the stability in the energy sense can be ensured when the appropriate algorithmic adjustments are made. This was shown by Simo and Tarnow³ for nonlinear elasticity, such that the “algorithmic energy” is conserved when the constitutive (material) evaluation is made appropriately. This algorithmic energy balance is in terms of the discretized nodal velocities and displacements. For a closed mechanical system

$$\mathbf{v}_{n+1}^T \mathbf{M} \mathbf{v}_{n+1} + U(\mathbf{u}_{n+1}) - \mathbf{v}_n^T \mathbf{M} \mathbf{v}_n - U(\mathbf{u}_n) = 0 \quad (1)$$

where \mathbf{v}_n is the vector of nodal velocities at time t_n ,

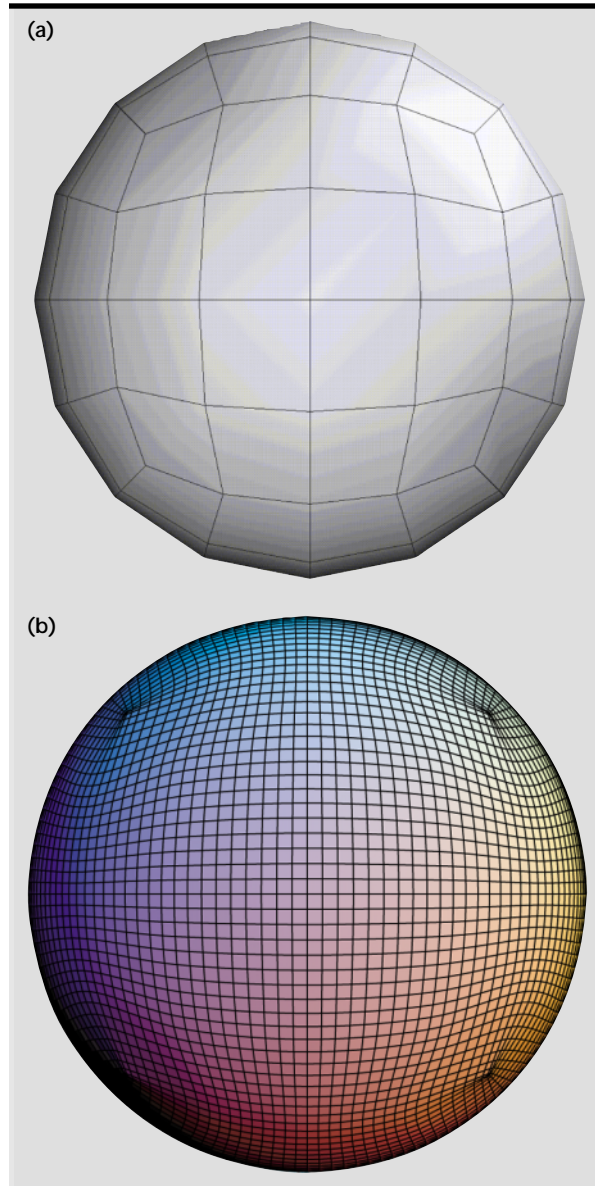


Figure 3. (a) Finite element model of sphere. (b) Smoothed version of faceted finite element mesh shown in (a).

\mathbf{u}_n is the vector of nodal displacements at time t_n , \mathbf{M} is the system mass matrix and U is some given strain energy function. **Equation 1** balances the energy at time t_n and t_{n+1} such that velocity and

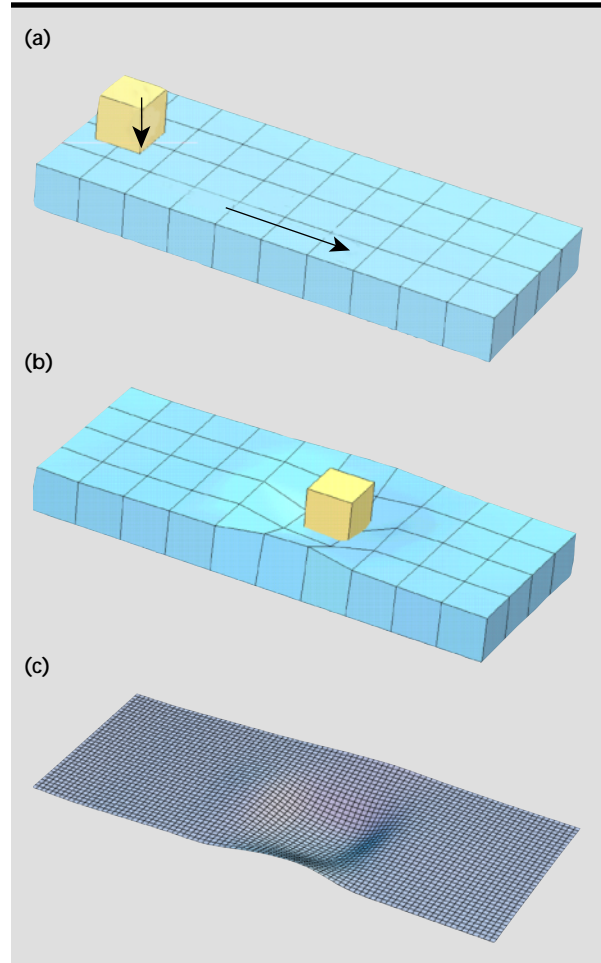


Figure 4. (a) Block loaded vertically and subsequently dragged along surface. (b) Final configuration. (c) Smoothed representation of surface.

Table 2. Energy norm results for block.

Iteration #	New method	Old method
1	2.0144×10^1	2.1177×10^1
2	4.5219×10^{-3}	-4.2748×10^1
3	5.4617×10^{-7}	-3.2830×10^1
4	1.3365×10^{-14}	5.1037×10^3
5	1.0987×10^{-28}	2.1842×10^2
6		1.3931×10^2
7		1.0540×10^2
8		8.4054×10^1
9		8.5236
10		3.3749×10^1 (diverged)

displacements are bounded and stability is ensured. In our work we generalize the energy conservation method to include coupled rigid/flexible body systems and contact with friction.

Ultimately, materials with plasticity will be included. With these effects, the new algorithmic balance is given by

$$\begin{aligned} \mathbf{v}_{n+1} \mathbf{M} \mathbf{v}_{n+1} + \mathbf{w}_{n+1} \mathbf{I} \mathbf{w}_{n+1} + U(\mathbf{u}_{n+1}) \\ - \mathbf{v}_n \mathbf{M} \mathbf{v}_n - \mathbf{w}_n \mathbf{I} \mathbf{w}_n - U(\mathbf{u}_n) = -D \leq 0 \end{aligned} \quad (2)$$

where \mathbf{I} is the rotational inertia of a rigid part, \mathbf{w} is the rigid body's rotational velocity and D is a positive amount of dissipation due to friction. We call this the algorithmic energy-consistent method. This new method is an extension of previous work.⁴⁻⁵

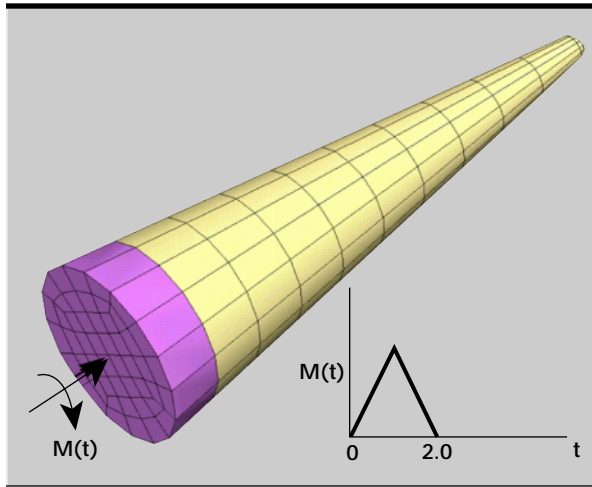


Figure 5. A pulse moment applied to a top with a rigid base and flexible cone.

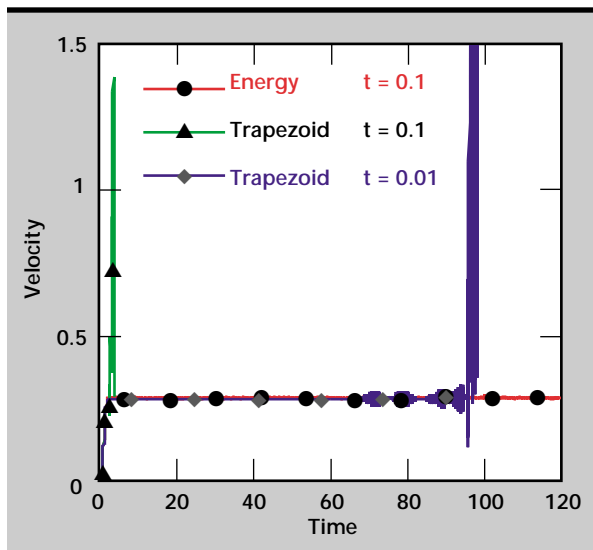


Figure 6. Velocity magnitude of a point near the tip of the top.

In the first example (Fig. 5), a pulse moment is applied at the base of rigid/flexible top. The base is a rigid material and the cone is flexible. The top is simulated using the new energy-consistent method and the classical trapezoid method. The resulting nodal velocity of a point near the top's tip is shown in Fig. 6. From Fig. 6, the trapezoid method is seen to become unstable for both the large ($\Delta t = 0.1$) and small ($\Delta t = 0.01$) time steps. The new energy method remains stable (independent of time step) as guaranteed by Eqs. 1 and 2.

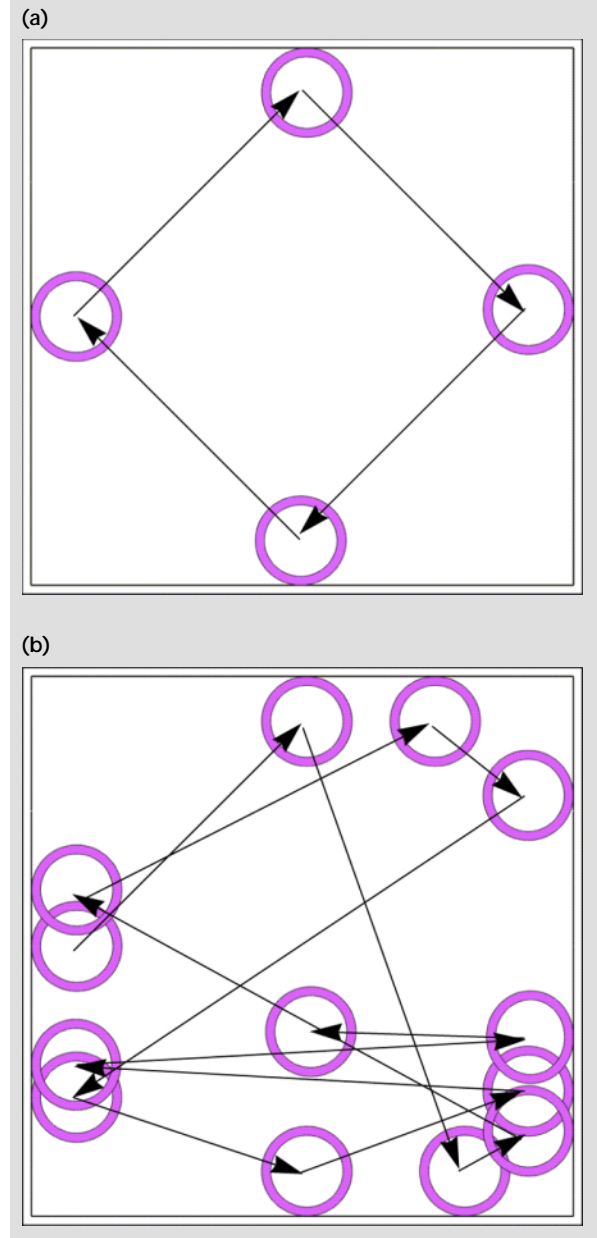


Figure 7. Flexible ring within a fixed rigid box. The ring is initially against the left wall and is given an initial velocity at 45° from the vertical. (a) The displacement pattern resulting from the new energy method. (b) The pattern resulting from the old trapezoid method.

In the next example (**Fig. 7**), a stiff flexible ring placed within a fixed rigid box is given an initial velocity at a 45° angle from the rigid wall. With the energy-consistent method, the ring bounces within the box in the expected way, returning almost exactly back to its original position in a given cycle (**Fig 7a**). With Newmark's trapezoid rule, the ring

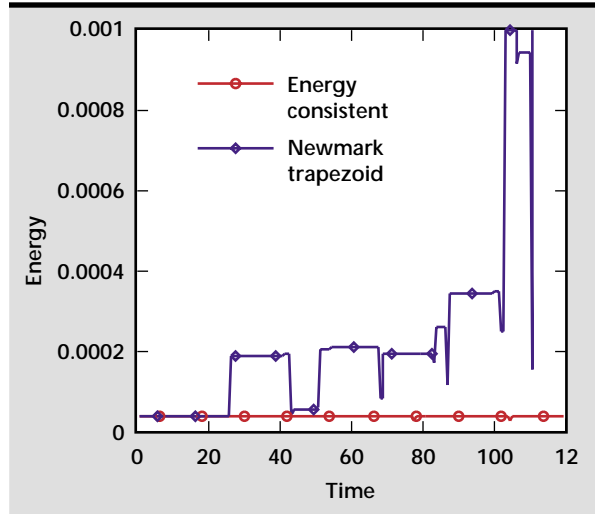


Figure 8. Algorithmic energy vs. time for the ring in box.

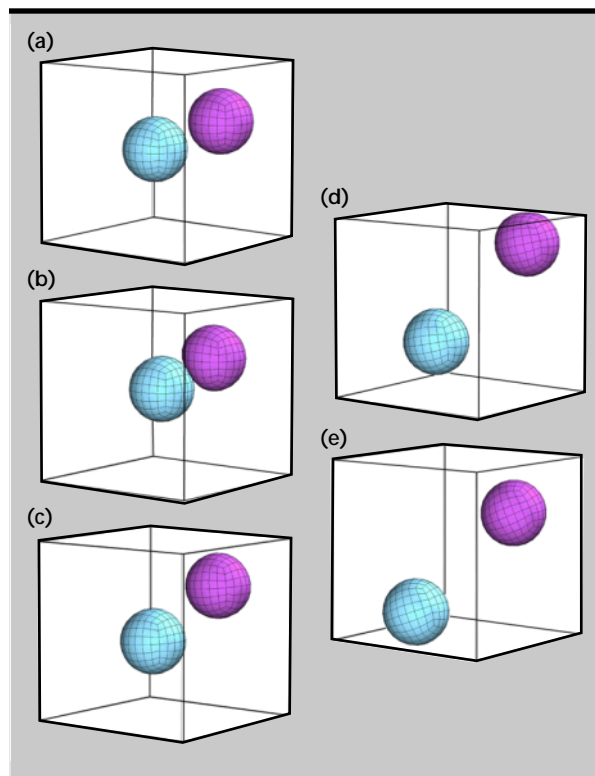


Figure 9. Sequence of configurations of flexible balls bouncing within a rigid box.

picks up energy at the boundary and eventually becomes unstable (**Fig. 7b**). The algorithmic energy is plotted versus time in **Fig. 8** for the ring in box.

In the final example, the effects of frictional dissipation are demonstrated. In **Fig. 9**, two flexible balls are given an initial velocity within a rigid (unfixed) box. The balls then bounce around within the rigid box as shown in the sequence in **Fig. 9**. Since friction is included the new energy-consistent method dissipates algorithmic energy as shown in **Fig. 10**. The trapezoid rule gains energy despite the friction.

Future Work

We have demonstrated the results of significantly more robust implicit finite element algorithms. Consequently, we are able to solve many new problems that were previously intractable.

Currently we are in the process of adding friction to the contact smoothing algorithm and adding the logic for determining locations in a mesh where smoothing is to be neglected. Furthermore, we are developing a complementary solver for computing the Lagrange multiplier contact forces more efficiently than the current augmented Lagrange method used in NIKE3D.

For implicit dynamics, we anticipate adding the dissipation effects of plasticity. We would also like to look at alternative conserving (or nearly conserving) algorithms that are simpler to implement.

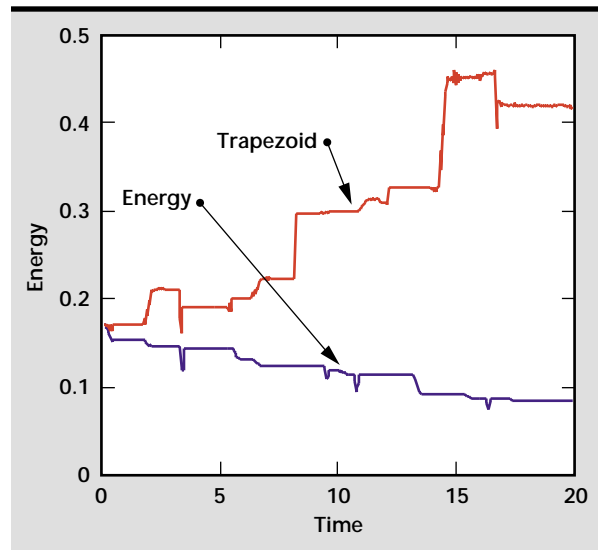



Figure 10. Algorithmic energy vs. time for the balls in box with friction.

References

1. M. Puso (to appear), "A highly efficient enhanced assumed strain hexahedral element," *IJNME*.
2. S. Wolfram (1993), "Mathematica: a system for doing mathematics by computer," Addison-Wesley Publishing Company Inc., Reading, Massachusetts.
3. J. Simo and N. Tarnow (1992), "The discrete energy-momentum method. Conserving algorithms for nonlinear elastodynamics," *ZAMP*, **43**.
4. M. Puso and E. Zywicz (1998), "Energy conserving rigid body contact methods," *ICES98 Proc*.
5. E. Zywicz and M. Puso (1999), "A general conjugate-gradient based predictor-corrector solver for explicit finite element contact," *Int. J. Numer. Meth. Engng.* **44**. 

ost-Processing and Data Management Software Development for Computational Mechanics

Douglas E. Speck
*Computer Applications Sciences and Engineering
 Computation Organization*

The introduction of the Mili I/O library to Lawrence Livermore National Laboratory (LLNL)'s Methods Development Group code suite, and an increased workforce, have spurred the release of a new GRIZ post-processor. These and other developments mean significant increases in capability and flexibility for users. New tools have been introduced to formalize and improve our software development process.

Introduction

In FY-99 we made considerable progress in the post-processing and data management areas of computational mechanics at LLNL. The unifying theme of our accomplishments in these areas is greatly improved flexibility. This leap forward was enabled and driven by the introduction of Mili (Mesh I/O Library), a library for reading and writing self-defining simulation databases. With our finite element post-processor, GRIZ, we completed a major redevelopment effort to accommodate the flexible data content permitted in Mili file families.

As a first step in integrating Mili into the analysis codes, code developers implemented modules in DYNA3D/ParaDyn and NIKE3D to write out the current collection of state variables to a Mili file family. Enhanced flexibility for post-processing extended to the parallel computing domain with the implementation of a prototype parallel rendering of GRIZ.

A second theme is improvement and formalization of our software development process. Building on an existing practice of maintaining source code in a repository accessed only by version control software, we added a web-based application for submission and management of software change requests and trouble reports. Additionally, a regression test suite was developed for GRIZ to automate the laborious process of validation testing new releases.

Progress

Mili Version 1.0

Version 1.0 of the Mili library was released to developers simultaneously with GRIZ 4.0, the new version of GRIZ that reads the Mili database. Mili accomplishes its primary goal of creating self-defining simulation databases by letting applications define formats that document the contents of state records to be written during a simulation. An application that reads the simulation database first reads the format descriptions to learn how to interpret the state data in the database.

Mili uses a conceptual model of an unstructured finite-element mesh as a collection of mesh objects (such as nodes or elements) for which simulation state data exists that an application may need to save to disk. Mesh objects with like topological and physical characteristics are grouped together in mesh object classes, each of which is identified by a unique class name. For example, hexahedral elements in a mesh might belong to a class called "brick." The application creates as many mesh object classes as necessary to fully define the mesh for data output.

As like objects make up a class, so can multiple classes be derived from the same superclass. Mili defines a set of superclasses that provide the basic semantics for each class created by the application.

Put another way, a class is a named instance of a superclass, and one of the parameters specified by the application when a class is created is its superclass. **Table 1** identifies the superclasses currently defined by Mili.

The reason for this superclass>class>object hierarchy is flexibility. From the superclass, an application knows very basic information about the class and therefore how to operate on it generally. For example, both **m_quad** and **m_tet** imply element mesh objects with four nodal connectivities, but a **m_tet** element is inherently 3-D and has obvious computational or rendering requirements associated with it, whereas a **m_quad** element can be either 2- or 3-D and has different associated requirements. With superclasses, Mili provides a logical shorthand for communicating basic information about groups of objects.

By permitting (demanding) applications to create classes, Mili allows specialization within a superclass. The superclass conveys common characteristics and the class confers some amount of uniqueness. This is useful when objects that might be identical at the superclass level are in fact modeled differently and mean something different to the analyst. For example, DYNA3D supports both thick shell elements and hexahedral volume elements, both with eight nodal connectivities. For writing data to the old TAURUS plot database, DYNA3D must combine both types of elements into that database's single collection of eight-noded elements. In post-processing, any original distinguishing information is lost and the analyst must know *a priori* which "bricks" are really bricks and which were thick shells in the analysis. Under Mili, DYNA3D can

create separate classes, each derived from superclass **m_hex** but having different class names, to allow each type of element to be explicitly identified in the output and subsequent post-processing.

Additional flexibility is available through Mili because the objects of a particular class can be subdivided on output for customizing the set of state variables being written. The most natural example of this would be a division of the elements within a class along material lines, since differing material models in the analysis would confer different sets of state variables for the participating subsets of elements. Finally, Mili also permits applications to define multiple state record formats in a single database. This opens the door to varying the amount of data being written dependent upon activity in the simulation.

GRIZ Version 4.0

The flexibility permitted in a Mili database represents a great change from the previous TAURUS plot database, and large parts of GRIZ were re-written (or invented) to accommodate the new features. GRIZ version 4.0 (GRIZ4) is the outcome of this redevelopment effort. The goal in producing GRIZ4 was to have GRIZ change outwardly as little as possible while accommodating the potential variability in a Mili database. Most of the command-set changes were made to accommodate the variable nature of database contents under Mili. For instance, commands that formerly required a parameter such as "node," "brick," or "shell" to indicate the object type of a numeric identifier now require a class name instead.

Visually, there are few changes in GRIZ4. Most apparent is the replacement of the previous "Result" pulldown menu with two new menus, "Derived" and "Primal." The Derived menu essentially holds the contents of the old Result menu, but the menu contents adapt to display only results that can be derived from the contents of the current database. The Primal menu allows access to all results written into the current database, without any derivation or transformation applied.

Through the Primal menu, GRIZ4 offers access to global mesh results and material results in both Mili and TAURUS databases. These data have been written to the TAURUS database for some time but were not previously accessed by GRIZ.

GRIZ4 does not support data from all Mili superclasses equally. There is no support for **m_wedge** or **m_pyramid** elements in 3-D rendering. Also, Mili's **m_unit** superclass, which by design has no semantics

Table 1. Mili object superclasses.

Class name	Node quantity	Object type
m_unit	N/A	Arbitrary
m_node	1	Node
m_truss	2	Truss
m_beam	3	Oriented beam
m_tri	3	Triangle
m_quad	4	Quadrilateral or shell
m_tet	4	Tetrahedron
m_pyramid	5	Pyramid
m_wedge	6	Wedge or prism
m_hex	8	Hexahedron or brick
m_mat	N/A	Material
m_mesh	N/A	Mesh

associated with it, cannot be related by GRIZ4 to the mesh in any way that is guaranteed to be generally useful for rendering. However, data from every class in a database, regardless of its superclass, can be read by GRIZ4 and plotted as a time history curve, so GRIZ4 has some utility even for object types it doesn't fully support.

Aside from the new adaptive menus, the only other visual change in GRIZ4's graphical user interface is on the Utility Panel. There, users now have the option of assigning any node or element class to any mouse button for picking operations, instead of the previous option to assign the middle mouse button to pick either shells or beams.

GRIZ's time history plotting capability was completely reworked for GRIZ4. Although the old "timhis" command is still supported, the new "plot" command provides users with much more capability and flexibility. Patterned after the THUG time history plotter's "plot" command, GRIZ4's "plot" accepts the specification of multiple results and/or mesh objects, generating a curve for every result/object pair supported by the database. GRIZ4's "plot" command syntax also permits parametric plotting of one result versus another result from the same mesh object (multiple objects allowed) with time as an implicit parameter. Additional development is planned, as the old legend is inadequate to describe the complexity now possible in time series plots.

Although GRIZ4 was developed primarily with the goal of integrating Mili into it, we knew from the outset that GRIZ4 would need to read both Mili and TAURUS databases, and considering developments under ASCI there was potentially a strong need to support additional I/O library interfaces as well. To accommodate multiple I/O interfaces, the GRIZ I/O (GIO) layer was implemented to abstract details of any particular I/O library and provide a common high-level internal interface to I/O functions. To interface to a given I/O library, a "driver" is written which provides the correct function interfaces for GIO but calls the desired I/O library internally. GRIZ4 has drivers for Mili and the TAURUS plot databases built in; other drivers can be added as dynamically-linked shared objects.

GRIZ4 Parallel Rendering Prototype

Pre- and post-processing are the two major bottlenecks facing code users in the parallel domain. This year we took a significant step to ease the post-processing bottleneck by developing a prototype parallel renderer in GRIZ4. We implemented a master/slave approach in which the

master processor manages the user interface and the slaves perform the parallel rendering. In the rendering algorithm, each processor renders its local mesh partition (determined by the domain decomposition from the analysis phase) followed by a global reduction of pixel red/green/blue values from all processors' images based on pixel Z (depth) values. In the final composited image, each pixel's color is taken from that initial image for which the depth value of the pixel was smallest (the pixel closest to the view eyepoint).

The parallel implementation was not intended as a production-ready solution but as a first step and baseline for experimentation and further development. Indeed, in using the domain decomposition from the analysis phase, we are almost assured of poor load balancing for rendering because only those local domains that have visible elements facing the view eyepoint can contribute pixels to the final image—the rendering effort on all other processors is wasted. Nevertheless, this was an attractive approach because of its simplicity and the fact that it offered the opportunity for parallel speedup while using the same serial rendering library (OpenGL) GRIZ already relies upon.

The parallel implementation must use offscreen rendering to memory, since the compute nodes of a parallel computer lack displays to render to. This has a beneficial by-product since it opens the door to "batch-mode" offscreen serial rendering as well, a long-sought capability for GRIZ.

Parallel-rendering GRIZ4 is still very much in the experimental stage. One of our priorities for the new fiscal year is to evaluate and profile the performance of this initial approach across an array of mesh sizes and processor quantities, then set the direction for a follow-on implementation leading to a production solution.

Combiner/Splitter Utility

An identified requirement associated with parallel processing is the need to remap the parallel outputs from an analysis run onto a different quantity of processors for subsequent post-processing or an analysis restart. In a similar vein, ParaDyn analysts need to combine *n* parallel Mili databases into one for serial post-processing. We met these needs by developing a combiner/splitter utility. Developed as a callable library with a driver program to provide the command-line interface, this utility will read multiple parallel Mili or TAURUS databases, plus the analysis partition file, and combine them (with translation, in the case of TAURUS) into a single Mili

database. The splitter code reads a second partition file for the target distribution of data and splits a single Mili database into the required number of parallel databases.

Web-Based Change-Request Tracking

In the past year, several factors combined to elevate the visibility of our software development process and the need for formalization. Fueled by ASCI funding, the growth in the number of code developers participating in graphics and data management software development highlighted the need for standardized conventions in our process and the use of software engineering tools. The *de facto* process arising from the old “one code, one code developer” environment is inadequate for a team of developers.

Taking advantage of LLNL’s Computation Directorate’s Software Technology Center, we requested a SCRTrack server to provide web-based management of software change requests (SCRs) and trouble reports. SCRTrack provides “cradle-to-grave” maintenance of SCRs on a product-by-product basis with support for a comprehensive set of user roles and SCR status conditions, plus automatic email notification of SCR state transitions. GRIZ4, Mili, and the GRIZ User Manual have been entered as products into the server, with plans for other codes to follow.

Regression Testing

Another aspect of the software process is quality assurance. One requirement associated with the release of GRIZ4 was the need to validate results against those from GRIZ version 2 (GRIZ2). This requirement manifests itself two ways. First, GRIZ4 must produce the same results as GRIZ2 when reading the same TAURUS database. Second, GRIZ4 reading a Mili database must produce the same results as GRIZ2 (or GRIZ4) reading results from the same analysis written out in a TAURUS database.

We developed a test script, written in the Python scripting language, that uses a standard set of DYNA and NIKE test problems and a collection of associated GRIZ command scripts to provide a basic validation of GRIZ4 with a comprehensive set of mesh configurations. The testing is not exhaustive in exercising all possible GRIZ capabilities, but it does thoroughly exercise data access and result computations and provides a baseline upon which to build additional testing capabilities. This addition to the GRIZ development process is valuable since it offers a great productivity boost for releasing GRIZ updates.

GRIZ4 ExodusII Driver

During the year, collaborators at Los Alamos National Laboratory (LANL) requested GRIZ modifications to read the Exodus II databases generated by Sandia’s Pronto analysis code. With GIO developed for such a purpose, GRIZ4 is ideally suited for this effort. By year’s end, GRIZ4 supported ExodusII, and it is currently in use by Engineering Services at LANL.

ASCI DMF Integration

Another I/O library compatibility requirement arises out of the ASCI program and its Data Models and Formats Application Programming Interface (DMFAPI), currently in active development. One of our goals under ASCI is to support DMFAPI in our “ASCI” analysis codes and in the GRIZ post-processor. GRIZ4’s GIO layer again offers a natural entry point for integrating the DMFAPI, but ParaDyn has no equivalent abstraction layer in its output code. Our chosen approach for ParaDyn, which follows a pattern established at Sandia and at LLNL, is to layer our current API (Mili) on top of the DMFAPI. This allows us to write out ASCI format databases with few if any source changes in ParaDyn. Essentially, only a re-link is required, replacing the Mili library with a new one that consists of the DMF library and a set of wrapper functions that conform to the Mili API.

The DMFAPI uses a very general data model with a strong, abstract mathematical foundation. It represents a significant departure from other I/O library APIs. We have layered Mili’s output calls over DMFAPI to enable ParaDyn to write out a DMFAPI database. The DMFAPI is still evolving, and with continued effort, its capabilities will be supported in other codes.

Future Work

Mili enables many changes, so it emerges as a natural focal point for additional development. Mili’s design is such that supporting additional superclasses is straightforward, and one such superclass already identified for inclusion is a “surface” or “face list” superclass. A surface object would be an arbitrary set of 3-D element faces, quad elements, or triangular elements that have special significance in the generating application. For example, a surface might be defined where a boundary condition is being applied in a simulation, or it might be a contact surface for which information exists in the simulation not found on all elements in general.

Another necessary capability that code developers have identified is restart management of the database; specifically, the ability to overwrite state records at an arbitrary point in the simulation. Our analysis codes implement slightly different approaches to this, so a generalized approach is needed.


The planned restart support will be impacted by a more recent requirement for Mili. An inter-code “link” file capability developed for the analysis codes is now scheduled to be re-expressed under Mili. This file was originally developed to convey state information between NIKE3D and DYNA3D. It is also now being used as a “lightweight” DYNA3D restart file. Implementing a Mili version of the link/stress-restart file imparts greater flexibility in its use (for instance, building on GRIZ4’s Mili compatibility, it opens the door to visualizing the link file) and provides an organized framework for ongoing extension of the file’s contents.

As mentioned above, GRIZ4’s prototype parallel rendering capability is scheduled to receive attention

in FY-01. First we will characterize the prototype’s performance over an array of problem sizes and processor quantities. This is expected to be followed by profiling to identify where bottlenecks lie in the current approach. These efforts will support a subsequent re-design of GRIZ4’s parallel rendering software.

Our software process improvement activities will continue into the new fiscal year. We have two priorities in this domain. First, we will expand use of SCRTrack to include other codes and user manuals. Second, we plan to develop software scripts which will interact with our source repository during release builds and automate versioning of application binaries (specifically GRIZ, but the motivation applies to all our codes).

Acknowledgments

The author acknowledges the work of Y. Lee, E. Pierce, V. Castillo, O. Simek, and M. Miller. 

Linear Equation Solvers for Parallel Computational Mechanics

Robert M. Ferencz
*Defense Technologies Engineering Division
Mechanical Engineering*

Chandrika Kamath
*Center for Applied Scientific Computing
Computation Organization*

Linear equation solvers present key performance issues for simulation codes using implicit time integration. This article summarizes our efforts to characterize solver methodologies suitable for inclusion in a new parallel implicit finite-element code.

Introduction

The Accelerated Strategic Computing Initiative (ASCI) program at Lawrence Livermore National Laboratory (LLNL) is tasking LLNL's Methods Development Group with the creation of a parallel implicit mechanics code. This effort builds upon our longstanding expertise in serial implicit finite-element codes for stress analysis (NIKE) and thermal analysis (TOPAZ), and will complement its ongoing development of the parallel *explicit* capability of the PARADYN code.

For present purposes, the fundamental difference between implicit and explicit simulation codes is that implicit codes use much larger step sizes for time integration. This makes implicit techniques suitable for modeling steady-state and low-frequency response, but this ability requires the solution of a set of linear equations at least once per time step. For the nonlinear materials and large-deformation phenomena typically modeled with NIKE, multiple linear solves are required at each time step. Hence, solving linear equations almost always represents the single largest share—at least 50%—of the overall computational cost in 3-D implicit simulations.

Linear equation solvers are typically divided into two classes: direct and iterative. Direct methods are known for their robustness and predictable number of operations. However, they require a global data structure that grows rapidly with problem size. Further, the algorithms used in direct solvers present challenges for large-scale parallelism.

Iterative methods typically use a much smaller data structure that can be readily distributed for large-scale parallelism. Yet they often perform poorly on problems exhibiting characteristics quite common in our simulations: material softening and damage, material anisotropy, mesh refinement to capture local effects, combined bending and membrane response of shells, and near-singularities as buckling behavior is approached. In addition to their memory and/or time cost, linear equation solvers can add significant complexity to the software architecture needed to support large-scale parallel computers. So, in designing the new parallel implicit code we not only need to incorporate the experience of PARADYN's development, but we must also extend it to address these added issues. In preparation for that design phase we have been performing studies of two linear solvers.

Progress

PSPASES Sparse Direct Solver

This solver was developed in the research group of Profs. V. Kumar and G. Karypis at the University of Minnesota.¹ We have previously interacted with this academic team that created the graph partitioning utility METIS used by PARADYN as part of its overall domain decomposition preprocessing. (The METIS package is also used by many other parallel applications developed throughout the world.) This same partitioning technology has now been applied to the

issue of equation reordering for efficient sparse direct equation solving on distributed memory machines. The PSPASES package solves symmetric, positive-definite systems of equations and must use a collection of 2^N processors, where N is an integer.

We are evaluating this solver technology and monitoring its evolution. Last year we created a simple prototype of a “master/slave” architecture. The first processor is essentially running the serial NIKE code: it performs all I/O, computes the finite-element matrices and assembles the global system of linear equations. At that time, the remaining $2^N - 1$ processors are sent portions of the global equation system, and then all processors execute PSPASES to reorder and solve the system. This solution is then consolidated on the first processor where the remainder of NIKE’s serial algorithms continue execution.

Obviously such a prototype is not a long-term approach to creating a production capability. But it has allowed us to exercise PSPASES within the context of a nonlinear finite-element code and provide feedback to its developers. For example, we identified and documented an anomaly in total memory use that occurred when multiple pairs of matrix reordering and factorization were requested in sequence. This is a fundamental requirement for nonlinear finite-element codes, but one not tested by the typical “test driver” that simply reads in a pre-computed matrix and does a single set of reordering/factorization/backsolve operations. With these data the developers were able to locate and correct the programming error, and memory usage has been verified to be essentially constant under the circumstances described. A growth in factorization CPU time has been documented for some situations and that information has also been passed to the developers.

Figure 1 shows a mesh used for some of the performance benchmarking. It is a subset of an overall model created by P. Raboin and S. Creighton for the Modal Analysis Correlation Experiment being undertaken jointly by LLNL, Sandia National Laboratories and the United Kingdom’s AWE. This particular mesh leads to a linear system having 250,007 degrees of freedom. The factorized form of the matrix contains 180×10^6 nonzero entries and requires 330×10^9 floating-point operations to compute.

Figure 2 shows timing data for this problem from runs on the ASCI “Blue Pacific” IBM SP system. The \log_2 of wall time for matrix factorization is plotted versus \log_2 of the number of processors used. The dashed line represents perfect scaling: the timing of each doubling of processors exactly halved the execution wall time. The PSPASES data show good

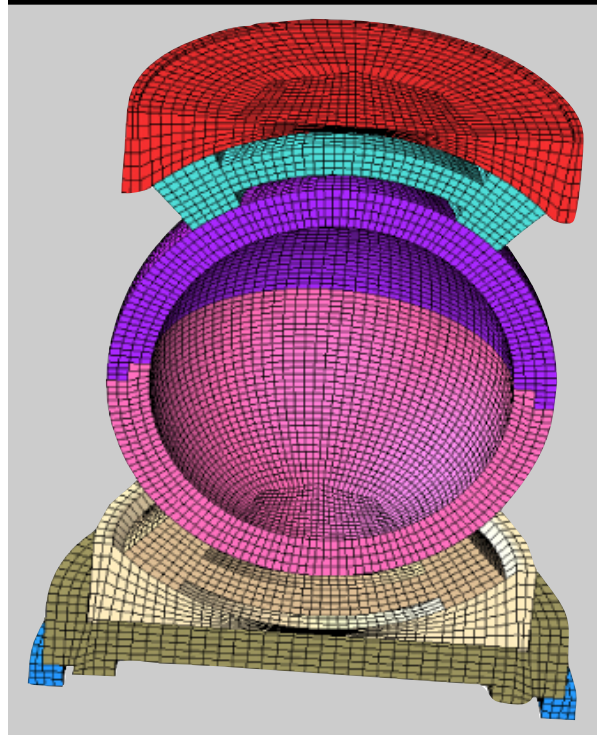


Figure 1. Mesh for a sub-assembly of the Modal Analysis Correlation Experiment (MACE). This mesh generates a system of 250,007 linear equations for stress analysis.

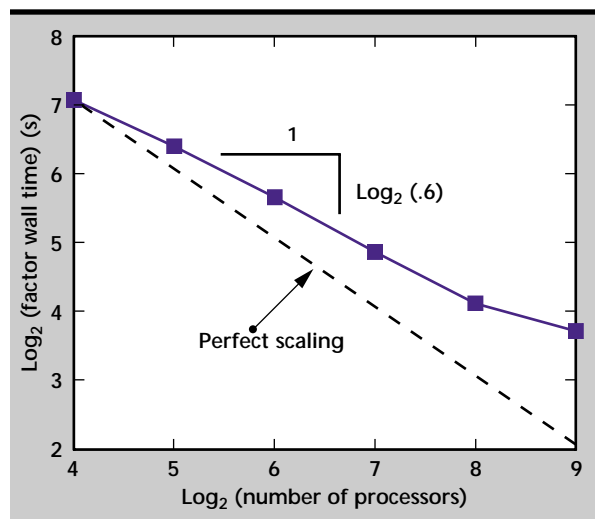


Figure 2. Timing data for the MACE benchmark run on the ASCI “Blue Pacific” IBM SP system.

scalability from 16 to 256 processors: each doubling of processors results in a 40% reduction in wall time. (Stated differently, each doubling of processors provides a 67% speed-up.) That the scalability suffers when pushed to 512 processors is not surprising, since that represents only 500 equations per processor and thus not many floating-point operations to amortize the increasing communication.

This work demonstrates that a distributed sparse direct equation solver can remain a viable methodology for implicit mechanics on at least hundreds of processors. This point had been in doubt—it is not uncommon to hear opinions that “direct solvers cannot scale beyond 8 or 16 processors.” While hundreds of processors represent only a fraction of an ASCI platform, it already would provide more than an order of magnitude increase in the size of implicit simulation we can perform in support of our engineering missions. The scalability of a direct solver on *thousands* of processors will be a longer-term issue, and motivates our examination of hybrid methods that combine iterative and direct equation solving technologies.

FETI

The Finite-Element Tearing and Interconnecting (FETI) method is a Lagrange-multiplier-based domain decomposition method. As a hybrid method, it has the potential of combining the best of both worlds, namely, the robustness of direct solvers and the scalability of iterative solvers as the number of processors is increased. Our work in this area has focused on developing a serial prototype of the FETI level 1 method,² with the aim of understanding the behavior of these methods on problems of interest to LLNL.

In the FETI method, the computational domain, Ω , is partitioned into N_s non-overlapping subdomains, $\Omega^{(s)}$. Lagrange multipliers, λ , are introduced at the subdomain interfaces to enforce the compatibility of the subdomain generalized displacements, $u^{(s)}$. As a result, the original linear system over Ω ,

$$Ku = f \quad (1)$$

where K , u , and f are the stiffness matrix, the displacement vector and the force vector, respectively, is reduced to the equivalent set of equations:

$$K^{(s)} u^{(s)} = f^{(s)} - B^{(s)T} \lambda \quad \text{for } s = 1, \dots, N_s \quad (2a)$$

$$\sum_{s=1}^{N_s} B^{(s)} u^{(s)} = 0 \quad (2b)$$

where, for each subdomain $\Omega^{(s)}$, the variables $K^{(s)}$, $u^{(s)}$, and $f^{(s)}$, denote the generalized stiffness matrix, the generalized displacements, and the generalized force vectors for that subdomain, respectively. The matrices $B^{(s)}$ are signed Boolean matrices that extract from a subdomain vector its signed contribution to the interface boundary of the problem.

If each of the subdomain stiffness matrices are non-singular, then we can solve for the subdomain displacements by first substituting $u^{(s)}$ from **Eq. 2a** into **Eq. 2b** and solving for λ , and then using **Eq. 2a** to solve for $u^{(s)}$. However, in practice, the domain decomposition process often generates one or more floating subdomains which do not have enough prescribed displacements to eliminate the local rigid body modes, resulting in a singular system in **Eq. 2a** for the corresponding subdomain. If this system is consistent, then the displacement $u^{(s)}$ for the floating subdomain $\Omega^{(s)}$ can be written in terms of λ as follows:

$$u^{(s)} = K^{(s)+} (f^{(s)} + B^{(s)T} \lambda) + R^{(s)} \alpha^{(s)} \quad (3)$$

where $K^{(s)+}$ is the pseudo-inverse of $K^{(s)}$, $R^{(s)}$ is the matrix of rigid body modes of $\Omega^{(s)}$, and $\alpha^{(s)}$ is a vector of unknowns that determines the contributions of each of the rigid body modes. By the definition of rigid body modes, the columns of $R^{(s)}$ form the basis of the null space of $K^{(s)}$. The introduction of the new unknowns $\alpha^{(s)}$ requires additional equations which can be obtained from the observation that for the singular equation to admit a solution, the right hand side must have no component in the null space of $K^{(s)}$, that is

$$R^{(s)T} (f^{(s)} + B^{(s)T} \lambda) = 0 \quad (4)$$

Combining **Eqs. 2a, 2b, 3, and 4** for all the domains in the problem, the FETI interface problem becomes

$$\begin{bmatrix} F_I & -G_I \\ -G_I^T & 0 \end{bmatrix} \begin{bmatrix} \lambda \\ \alpha \end{bmatrix} = \begin{bmatrix} d \\ -e \end{bmatrix} \quad (5)$$

where

$$\begin{aligned} F_I &= \sum_{s=1}^{N_s} B^{(s)} K^{(s)+} B^{(s)T} \\ G_I &= \begin{bmatrix} B^{(1)} R^{(1)} & B^{(2)} R^{(2)} & \dots & B^{(N_f)} R^{(N_f)} \end{bmatrix} \\ d &= \sum_{s=1}^{N_s} B^{(s)} K^{(s)+} f^{(s)}, \\ \alpha &= \begin{bmatrix} \alpha_1^T & \alpha_2^T & \dots & \alpha_{N_f}^T \end{bmatrix}^T, \\ e^{(s)} &= \begin{bmatrix} f^{(s)} R^{(s)} \end{bmatrix}^T, \quad s = 1, \dots, N_f \end{aligned}$$

where N_f is the number of floating subdomains in the problem. The matrix $K^{(s)+}$ reduces to $K^{(s)-1}$ for non-singular systems.

While the matrix F_I is symmetric positive-definite, the system matrix in **Eq. 5** is indefinite, and we cannot directly apply the conjugate gradient method to solve for the Lagrange multipliers. However, we

observe that **Eq. 5** is equivalent to solving the equality constraint problem

$$\underset{\lambda}{\text{Minimize}} \quad \Phi(\lambda) = \frac{1}{2} \lambda^T F_I \lambda - \lambda^T (G_I \alpha + d)$$

$$\text{Subject to } G_I^T \lambda = e$$

This implies that we can use the conjugate gradient method with projected gradient to solve **Eq. 5**, with F_I as the system matrix, d as the right hand side and λ as the unknowns. The conjugate gradient with projected gradient technique is identical to the conjugate gradient algorithm, but, in addition, it ensures that the constraint is satisfied at each iteration. If we choose the initial λ to satisfy the constraint, that is,

$$\lambda_0 = G_I (G_I^T G_I)^{-1} e$$

then the constraint is satisfied at each iteration if

$$G_I^T \lambda_k = 0$$

which can be ensured by projecting the direction vectors in the conjugate gradient method onto the null space of G_I^T , using the projector

$$I - G_I (G_I^T G_I)^{-1} G_I^T.$$

As with the standard conjugate gradient method, the performance of the basic conjugate gradient with projected gradient method can be improved through the use of preconditioning. Farhat and Roux³ suggest the following two preconditioners:

$$\text{Lumped: } F_I^{-1} = \sum_{s=1}^{N_s} B^{(s)} \begin{bmatrix} 0 & 0 \\ 0 & K_{bb}^{(s)} \end{bmatrix} B^{(s)T}$$

Dirichlet:

$$F_I^{-1} = \sum_{s=1}^{N_s} B^{(s)} \begin{bmatrix} 0 & 0 \\ 0 & K_{bb}^{(s)} - K_{ib}^{(s)T} K_{ii}^{(s)-1} K_{ib}^{(s)} \end{bmatrix} B^{(s)T}$$

where the matrix $K^{(s)}$ is partitioned as

$$K^{(s)} = \begin{bmatrix} K_{ii}^{(s)} & K_{ib}^{(s)} \\ K_{ib}^{(s)T} & K_{bb}^{(s)} \end{bmatrix}$$

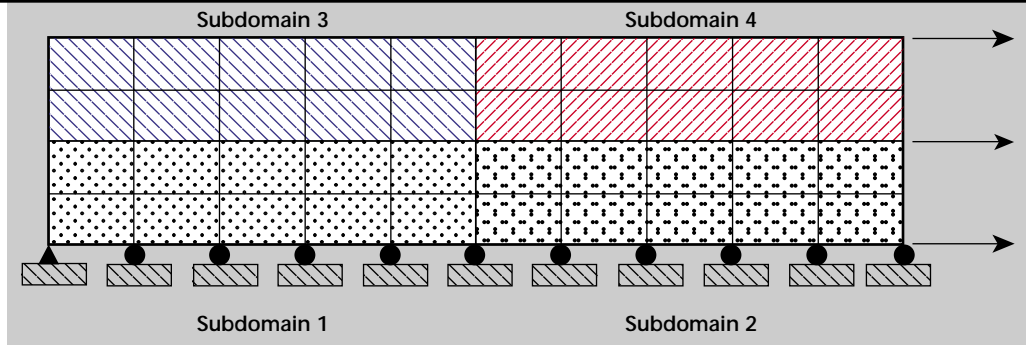
where the subscripts i and b indicate interior and boundary variables, respectively. Note that the Dirichlet preconditioner is a better approximation to the inverse of F_I , albeit a more expensive one.

We have implemented a serial version of the FETI level 1 method in Fortran 77. The current version of the code can handle cross-points in the decomposition as well as heterogeneous domains.⁴ The subdomain problems that arise in the application of the conjugate gradient with projected gradient technique are solved using a skyline solver. While this is not the most efficient direct solver, it does enable easy calculation of the pseudo-inverses and rigid body modes.²

To understand the behavior of the FETI method, we experimented with the problem in **Fig. 3**, which solves the equations of static equilibrium for a simple 2-D bar extension problem using quadrilateral elements. The problem has four subdomains, with a single cross-point where the four subdomains meet. Subdomains 2 and 3 are floating subdomains with a single rigid body mode each, while subdomain 4 has three rigid body modes. The iterations of the preconditioned conjugate gradient with projected gradient method are stopped when the two norm of the preconditioned projected residual is less than 10^{-8} . Further details on the implementation of the method are given elsewhere.⁵

Table 1 lists the iterations for the FETI method for our simple test problem as the properties of the subdomain are varied to represent problems of interest to LLNL. We observe that the number of iterations required for convergence, especially with the Dirichlet preconditioning, remains constant as the problem is made more ill-conditioned. Note that these are the iterations required to solve the interface problem using the preconditioned conjugate gradient with

Figure 3. Test problem for FETI level 1. The global problem has 94 equations, while the interface problems has 38 equations.



projected gradient method. Finally, to compare the performance of the FETI method with standard iterative schemes, we present in **Table 2** the results of solving the global system using the iterative solvers from Compaq's DXML™ solver library.⁶ This data shows that while standard iterative solvers may work in some problems, they are not always scalable as the problem becomes more ill-conditioned.

These early results demonstrate that the FETI technique shows promise of being a viable method

for solving ill-conditioned problems when the source of ill-conditioning arises from commonly occurring situations in LLNL simulations. The scalability of the FETI method on problems of a size of interest to ASCI has already been demonstrated⁷ on the ASCI option Red supercomputer. However, as outlined below, several additional experiments are necessary before we can say that the FETI method can solve difficult problems of interest in an efficient and effective way on massively parallel machines.

Table 1. Iteration counts for the FETI method on the test problem in Fig. 3. The iterations are for the solution of the interface problem using preconditioned conjugate gradient with projected gradient technique.

	No preconditioning	Lumped preconditioning	Dirichlet preconditioning
Problem 4 Homogeneous stiffness	22	13	10
Problem 4a Subdomain 4: 10× stiffer	25	13	10
Problem 4b Subdomain 4: 1000× stiffer	24	13	9
Problem 4c SD1: 100×, SD2: 10×, SD3: 1×, SD4: 1000×	34	17	10
Problem 4d Homogeneous, nearly incompressible elasticity	22	34	15
Problem 4d With re-orthogonalization	20	27	12

Table 2. Iteration counts for standard conjugate gradient techniques with various preconditioners applied to the global system obtained from the problem in Fig. 3.


	No preconditioning	Diagonal preconditioning	Polynomial preconditioning	ILU preconditioning
Problem 4a	94 (close to the solution)	67	48	25
Problem 4b	94 (not close to the solution)	86	62	34
Problem 4c	94	75	55	28
Problem 4d	No convergence	No convergence	No convergence	Preconditioned matrix not positive-definite
Tolerance: 10^{-8}				
Maximum iterations permitted: 94				
Order of polynomial preconditioner: 2				

Future Work

Our work with PSPASES will now be folded into the on-going design effort for LLNL's new parallel implicit mechanics code. By considering from the outset the requirements of both direct and iterative/hybrid equation solvers, the code will incorporate methodologies suitable for a broad range of simulations. The numerical robustness of the sparse direct solver will also be useful during the development of the code, as it will permit us to focus on other aspects of the code's overall behavior and performance.

Our work on the FETI level 1 method has so far focused on simple 2-D problems, where the subdomains are homogeneous, though the material properties may be different across subdomains. To ensure that this method is indeed a versatile one, we need to repeat our experiments on simple 3-D problems, and study the effects of heterogeneity when it occurs within a subdomain. We also plan to study the effects of changing the aspect ratio of the subdomains. The subdomain aspect ratio is an important factor in determining the convergence of the FETI method.³ If we use a mesh partitioning software such as METIS, it is not clear if it will provide subdomains with near-unity aspect ratios, or if additional work will be required, especially as the simulation evolves over time and the partitioning changes.^{8, 9}

References

1. M. Joshi, G. Karypis, V. Kumar, A. Gupta and F. Gustavson (1999), "PSPASES: An efficient and scalable parallel sparse direct solver," in *Proc. Int. Workshop on Frontiers of Parallel Numerical Computations and Applications*, Annapolis, Maryland, February.
2. C. Farhat and F. X. Roux (1991), "A method of finite element tearing and interconnecting and its parallel solution algorithm," *Int. J. Numer. Meth. Eng.*, Vol. **32**, pp. 1205–1227.
3. C. Farhat and F. X. Roux (1994), "Implicit parallel processing in structural mechanics," *Comp. Mech. Adv.*, Vol. **2**, No. 1, pp. 1–124, June.
4. D. Rixen and C. Farhat (1999), "A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems," *Int. J. Numer. Meth. Eng.*, Vol. **44**, pp. 489–516.
5. C. Kamath, "The FETI level 1 method: theory and implementation," LLNL Technical report, in preparation.
6. C. Kamath, R. Ho, and D. Manley (1994), "DXML: A high performance scientific subroutine library," *Digital Technical Journal*, Vol. **6**, No. 3, Summer.
7. M. Bharadwaj, D. Day, C. Farhat, M. Lesoinne, K. Pierson, and D. Rixen (1999), "Applications of the FETI method to ASCI problems: scalability results on one thousand processors and discussion of highly heterogeneous problems," Sandia Report, SAND99-0937, April.
8. C. Farhat, N. Maman, and G. Brown (1995), "Mesh partitioning for implicit computations via iterative domain decomposition: Impact and optimization of the sub-domain aspect ratio," *Int. J. Numer. Meth. Eng.*, Vol. **38**, pp. 989–1000.
9. D. Vanderstraeten, C. Farhat, P.S. Chen, R. Keunings, and O. Ozone (1996), "A retrofit based methodology for the fast generation and optimization of large-scale mesh partition: beyond the minimum interface size criterion," *Comp. Meth. Appl. Mech. and Eng.*, Vol. **133**, pp. 25–45. 



EIGER Preprocessor and User Interface

Robert M. Sharpe, Nathan J. Champagne, and Andrew J. Poggio
Defense Sciences Engineering Division
Electronics Engineering

Our goal is to facilitate the use of advanced computational electromagnetics (CEM) tools by application designers and analysts. Specifically, this entails developing an advanced preprocessor and graphical user interface (GUI) for frequency-domain electromagnetics codes (for example, EIGER). The preprocessor reads and interprets the mesh files that are output from commercial mesh generation packages, and assists with the decisions and assignments that are needed to perform the requisite CEM analysis. The complexity of the geometry often requires automated diagnostics to ensure proper connectivity and discretization of the problem. In addition, the wide variety of analysis options available in the new generation of CEM tools necessitates an interface that assists users with the decision process and problem setup so that designers may concentrate on their specific analysis rather than on the CEM issues.

Introduction

Modern software engineering methods enable analysis tools development with an unprecedented degree of flexibility in incorporating different formulations for advanced physics, and numerical methods for a single self-consistent solution. This in turn yields general-purpose tools, such as the EIGER¹⁻⁴ software suite, with an unparalleled range of applicability.

Current applications include advanced micro-electromechanical machines (MEMMs), microwave components (RF isolators and splitters), radar cross-section (RCS) analysis, broad-band antenna design, frequency-selective surface design, Navy

ship topside design, and phased array analysis and design. This wide variety of applications on one hand and the extensive set of features and options in the computational tools on the other hand yield a technology gap that must be bridged to successfully transition this technology to the various user communities both within and outside Lawrence Livermore National Laboratory (LLNL).

Progress

The graphical user-oriented interface (EIGER Build) is a principal part of the EIGER framework (**Fig. 1**) for frequency-domain electromagnetics analysis. It assists

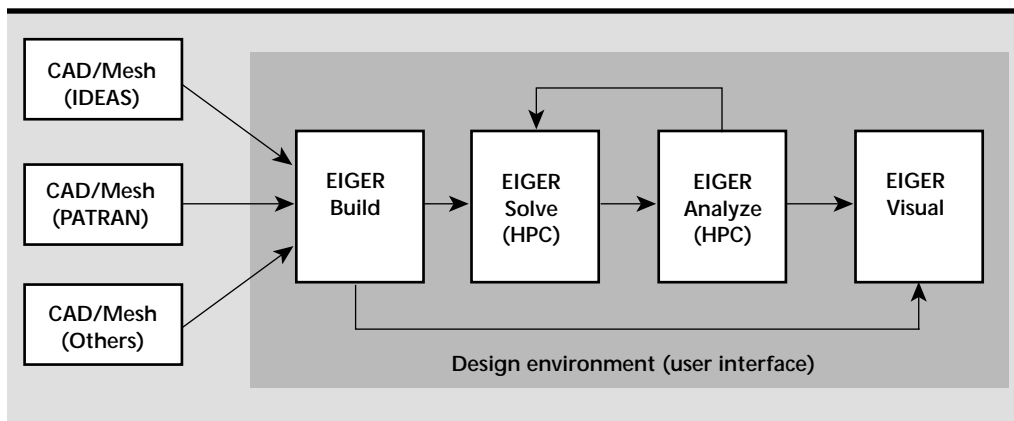


Figure 1. The four components of the EIGER Framework. EIGER Build is the graphical preprocessor; EIGER Solve is the physics engine; EIGER Analyze is the physics post-processor; and EIGER Visual is the graphical post-processor.

designers in the most time-consuming and error-prone portions of a CEM simulation—the problem setup and definition phase. Simulation files are created in EIGER Build based on a project concept.

The project contains all of the components needed for a CEM model, such as the geometry, the boundary conditions, and the excitations. The meshed geometry is created by a CAD/mesh program, such as PATRAN or I-DEAS, and then read into EIGER Build. The discretized geometry is visually inspected in EIGER Build for errors. If the mesh is satisfactory, CEM-specific information to the geometry is interactively applied to complete the simulation model. By storing the simulation model in a project file, any minor updates made in the CAD program will not adversely affect the project. This allows, for example, geometrical parameter studies to be done with minimal effort.

An EIGER Build project is composed of a set of databases. The entries in each are associated, by EIGER Build, with parts of the geometry or the model as a whole. The databases include:

1. visuals, for controlling the appearance of the geometry;
2. boundary conditions, for specifying the presence of electric and magnetic currents;
3. integral equations, for enforcing the selected boundary conditions;
4. materials, for augmenting the geometry information with electromagnetic properties;
5. plane waves and voltage sources, for exciting the model;
6. lumped loads, for representing ports;
7. spectral information, for picking the desired frequencies and modes;
8. far-field sampling points, for describing grids of angles at which far-field calculations are desired; and
9. near-field sampling points, for describing grids of points at which total field calculations are desired.

A sample input database form for a plane wave excitation is shown in **Fig. 2**.

There are several analyses that are performed to assist in geometry validation. The analyses are done automatically after a mesh is read into EIGER Build. However, these analyses can be redone via menu selections that produce visual results. The different analyses available are as follows:

Boundary edges. The surface elements in a mesh are analyzed for open boundaries. These are edges of any surface element that do not connect to another surface element. This collection of element edges can be thought of as the apertures. This analysis is used to look for unintended (or missing) apertures in a model.

Joint edges. The surface elements in a mesh are analyzed for multiply connected boundaries. These

Figure 2. Database input screen for a plane wave excitation.

PlaneWaveDef	
Name	Plane Wave
Region Id	1
Layer index in region	0
theta	0
phi	0
Re(Htheta)	0
Im(Htheta)	0
Re(Hphi)	1
Im(Hphi)	0

Reset Save

are edges of any surface element that connect to more than one other surface element. This analysis is used to look for unintended (or missing) contact sites.

Surface edges. The surface elements in a mesh are analyzed for singly connected edges. These are edges of any surface element that connect to exactly one other surface element. This collection of element edges can be thought of as the ordinary edges of the mesh. As such, its information is limited, but it does provide the complement of the other surface element analyses. This analysis is used as an eyeball aid while viewing other analyses.

Wires. The wire elements in a mesh are collected into a single group. This collection provides a simple way to view all the wires at once. The collection even includes wire elements that were not put into groups. This analysis is used to verify that all of the wires are present.

Free wire ends. The ends of wire elements in a mesh are analyzed for unattached ends. These are ends of any wire element that do not connect to another wire or surface element. This analysis is used to verify that all of the wires are properly attached to the geometry.

Junction wire ends. The ends of wire elements in a mesh are analyzed for ends touching surfaces. These are ends of any wire element that connect to

at least one surface element. This analysis is used to verify that all of the appropriate wires are properly attached to the surface geometry.

An example of an analysis result in EIGER Build is shown in **Fig. 3**. The meshed structure is a Navy patrol craft modeled with surface and wire elements. The particular analysis shown is “free wire ends,” which is used to determine if all the wires are attached properly. This is visually shown by crosses at the wire ends. A close up of a collection of wires near the bow is shown in **Fig. 4**. Upon closer inspection, it is determined that a wire that should be attached to the surface is not attached after all. If an electromagnetic simulation were run with this error, the results would not be correct, especially in the area surrounding this collection of wires.

Future Work

During the next phase of this project, we will incorporate volume finite elements into the EIGER software suite. This is a necessary step to treat geometries with a hybrid finite element/boundary element formulation. Additionally, higher-order interpolatory functions and elements will be addressed to further enhance the capabilities of the EIGER suite by increasing the accuracy and efficiency for more complicated models.

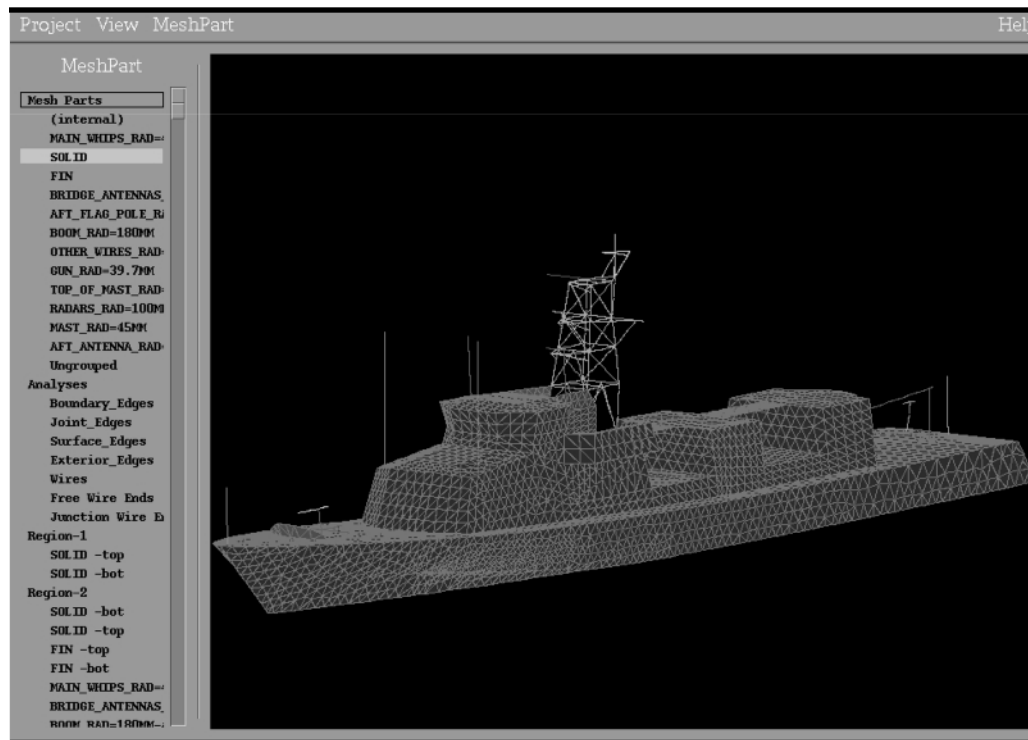
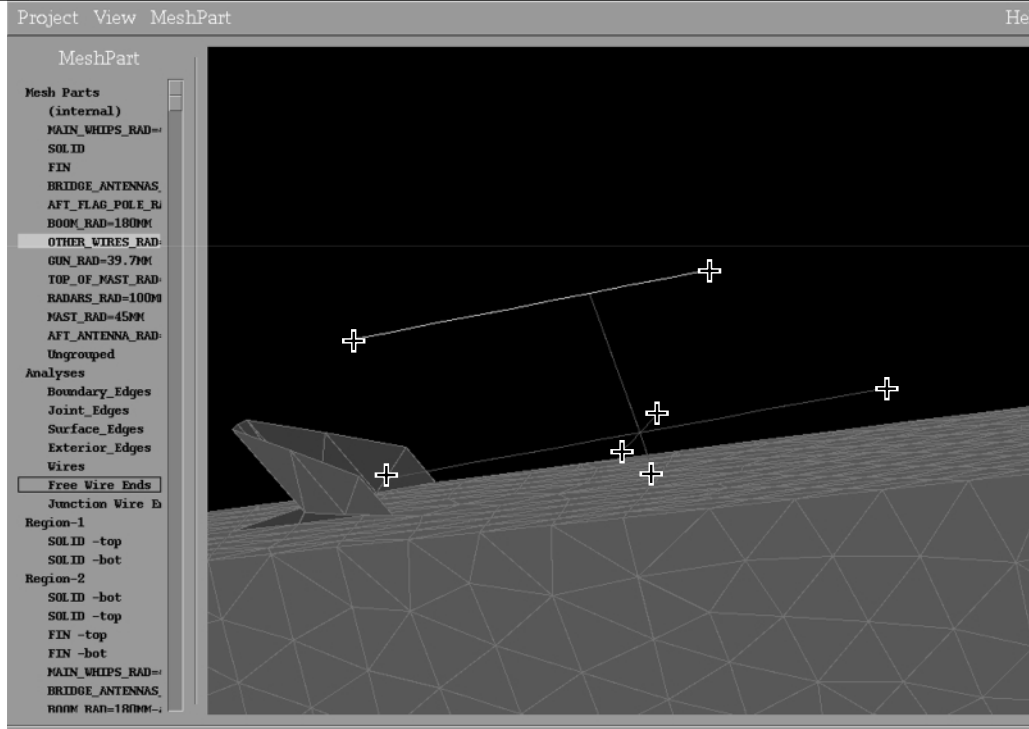


Figure 3. Screen shot of EIGER Build. The geometry is a Navy patrol craft. The crosses at the wire ends indicate that they are unattached.

Figure 4. Close-up of a collection of wires near the bow of the craft shown in Fig. 3. The crosses indicate that the wires are not attached to the craft surface.



References

1. R. M. Sharpe, J. B. Grant, N. J. Champagne, W. A. Johnson, R. E. Jorgenson, D. R. Wilton, W. J. Brown, and J. W. Rockway (1997), "EIGER: Electromagnetic Interactions GEneralized," *IEEE AP-S International Symposium and North American URSI Radio Science Meeting*, Montreal, Canada, July.
2. R. M. Sharpe, J. B. Grant, N. J. Champagne, W. A. Johnson, R. E. Jorgenson, D. R. Wilton, D. R. Jackson, J. W. Rockway, and C. W. Manry (1998), "Electromagnetic Interactions GEneralized (EIGER): Algorithm abstraction and HPC implementation," *Twenty-Ninth Plasmadynamics and Lasers Conference*, Albuquerque, New Mexico, June.
3. W. A. Johnson, R. E. Jorgenson, L. K. Warne, J. D. Kotulski, J. B. Grant, R. M. Sharpe, N. J. Champagne, D. R. Wilton, and D. R. Jackson (1998), "Our experiences with object oriented design, Fortran 90, and massively parallel computations," *IEEE AP-S International Symposium and North American URSI Radio Science Meeting*, Atlanta, Georgia, June.
4. W. A. Johnson, R. E. Jorgenson, L. K. Warne, J. D. Kotulski, J. B. Grant, R. M. Sharpe, D. R. Wilton, and D. R. Jackson (1999), "Our experiences with object oriented design, FORTRAN90, and massively parallel computations," *Progress In Electromagnetics Research Symposium Proceedings*, March 22-26, **1**, Taipei, Taiwan, p. 164.



Strength and Reliability of Brittle Materials

Robert A. Riddle
Defense Technologies Engineering Division
Mechanical Engineering

Choi K. Syn and Michael J. O'Brien
Manufacturing and Materials Engineering Division
Mechanical Engineering

Steve Duffy, Joe Palko, and Eric Baker
Connecticut Reserve Technologies

We have tested the strength and reliability of brittle materials using statistical fracture mechanics incorporating Weibull parameters into finite-element methods. Results are given for four-point bend bar tests on beryllium oxide.

Introduction

Components fabricated from brittle materials find wide use in numerous programs at Lawrence Livermore National Laboratory (LLNL). Examples include structural ceramics, windows and synchrotrons in particle accelerator technology, and optical components for lasers. The use of components of such materials in these applications requires knowledge of appropriate stress levels through a wide variety of loading scenarios.

Traditional methods of safety assessment work less well for brittle materials because of the larger scatter in strength values as determined by materials testing, and because the amount of data scatter increases as the component increases in size. Use of statistical fracture mechanics incorporating Weibull parameters will allow a rational, state-of-the-art, and probabilistic prediction of failure. Used in conjunction with finite-element methods, safe stress levels may be chosen based on the required reliability.

An important part of this effort is the incorporation of the state of the art in the design and application of advanced ceramic materials. The CARES (Ceramic Analysis and Reliability Evaluation for Structures) program from NASA Lewis represents a successful effort.¹ The CARES program takes a

description of the state of stress from brittle material and derives a probability of failure based on Weibull strength test parameters and estimates of the effects of the biaxiality of the stress state. Our effort incorporates the CARES type failure prediction within constitutive models for brittle materials within the LLNL finite-element codes.

Progress

Weibull Statistical Method for Failure Prediction

The Weibull method for statistical failure prediction is based on the concept of a distribution of flaws both in terms of size and orientation within the microstructure of a brittle material. Weibull described this method as a 'weakest link' description for the size effect on failures in solids.² Then, following the derivation of Robinson,³ for small volume elements where the stress can be assumed to be uniform, the probability of failure per unit volume is P_0 . The probability of survival, S , of a volume, V , of the material is given by

$$S = (1 - P_0)^V \quad (1)$$

for the simultaneous survival of small volume elements.

Taking the logarithm of each side of **Eq. 1** gives

$$\ln S = -V \ln(1 - P_o).$$

The risk of rupture is defined as $R = -\ln S$, which in terms of an infinitesimal element gives

$$dR = -\ln(1 - P_o) dV.$$

The term $-\ln(1 - P_o)$ is assumed to be a positive function depending on only the tensile stress $n(\sigma)$. Then

$$dR = n(\sigma) dV$$

and

$$R = \int_V n(\sigma) dV$$

where R is the risk of rupture integral. From the definition of R we get the probability of failure as

$$P_f = 1 - \exp[-\int_V n(\sigma) dV].$$

For the three parameter Weibull distribution

$$n(\sigma) = \left(\frac{\sigma - \sigma_u}{\sigma_0} \right)^m,$$

known as the general Weibull distribution function. In this distribution the quantity 'm' is unitless and is termed the shape factor or Weibull modulus. It is a measure of the variability of the material strength distribution. The quantity σ_0 is the Weibull scale parameter and has units of stress volume^(1/m). For a material component of unit volume, σ_0 is the expected value of strength at the 63.2% probability level. The quantity σ_u is the Weibull threshold parameter, and is the stress value below which no failures are expected to occur. For conservatism and mathematical convenience, the threshold stress is normally considered to be zero, and a simpler two-parameter Weibull model is used.

Weibull Prediction of Size Effect and Stress Distribution

Consider two similar components of the same material but of different volumes, V_1 and V_2 . If the stress state is uniform tension, the size effect in the predicted strength for a two-parameter Weibull failure model is:³

$$\frac{\sigma_1}{\sigma_2} = \left(\frac{V_2}{V_1} \right)^{\left(\frac{1}{m} \right)}$$

As an example of the size effect, consider the case where the volume ratio of the two components is 10, and the Weibull modulus of the material is 10, as would be typical for the high strength ceramic. Then the strength ratio is

$$\frac{\sigma_1}{\sigma_2} = \left(\frac{10}{1} \right)^{\left(\frac{1}{10} \right)} = 1.26.$$

The large component would be only 80% as strong as the small component.

A similar effect may be shown for size effect,³ where a component in tension would be predicted to have 73.5% of the strength of a component under pure bending.

These are the size and stress distribution effects in the simplest of circumstances. When 3-D combined stress states, and both volumetrically and surface distributed flaws are considered, as would be the case in most real applications, effects of size and stress distribution take a more complex form.⁴ It has been shown that some brittle materials show a definite size effect (glass cylinders under compressive loads), while others do not (cement paste cylinders). The appropriate use of a Weibull distribution to model a material's failure behavior depends on the details of the observed failure data.

Correlation of Fracture Toughness and Weibull Parameters

The fracture mechanics approach to failure prediction is deterministic, with failure predictions based on known flaw sizes and the material's resistance to crack growth, called the fracture toughness. Based on a given flaw geometry, and the far-field load application, the applied stress intensity factor may be calculated at the crack tip. When the applied stress intensity factor equals the critical stress intensity factor, or fracture toughness, crack propagation occurs.

By contrast, Weibull statistics predicts failure based on unknown flaw distributions and far-field stress fields. The results are predicted probabilities of crack propagation. It is possible that there are connections between the two methods.

One attempt to categorize the scatter for fracture toughness measurements⁵ could correlate the two methods. Fracture is treated as a weakest link phenomenon, while failure is treated in a probabilistic fashion. The fundamental expression for the probability of failure is given by

$$P_f = 1 - \exp[-\int_V \rho dV].$$

Here V is a specified volume, and ρ is a failure function per unit volume.

It would seem that to use the preceding equation the function ρ must be defined, or its characteristics must be inferred. This function must be dependent upon stress. As noted above, Anderson⁵ cites earlier work by Ritchie, Knott and Rice⁶ where a model was postulated that predicted cleavage failure when the stress ahead of a crack exceeds the failure stress of the material (σ_f) over a characteristic length. Later work by Curry and Knott⁷ used statistical arguments to show that a finite volume of material must be subjected to a stress field severe enough to fail, with a defect large enough to initiate cleavage.

Thus the volume stipulated in the preceding equation must be a region surrounding the crack tip. The size of the region is defined by the severity of the crack tip stress field and the inert strength of the material. The inert strength of a material is defined by ρ and because of this a specific form for ρ is never needed. The material tested will dictate what this function is. In a sense the material is the function. Test loads and boundary conditions serve as input and test data serves as output from the function. However, the crack tip stress field must be specified to derive the cumulative distribution functions (cdf) for K_{IC} (fracture toughness) and J_C (critical energy release rate). Since cleavage fracture is the assumed failure mechanism, stress fields associated with small scale yielding (SSY) must be used.

The function ρ is a function of the stress field near a crack tip. Using the work of Hutchinson⁷ the stress field can be shown to take the form

$$\sigma = f\left(\frac{J}{r}, \theta\right).$$

Anderson⁵ invokes the relationship between the J-Integral and the stress intensity factor K to obtain the expression

$$P_f = 1 - \exp\left\{-\left(\frac{K_{IC}}{\theta_K}\right)^4\right\}.$$

One implication of this relationship is that when a single large flaw under tensile loading is the dominant failure mode in a material, the Weibull modulus would be measured as 4.0. Other effects such as loading mode mixity, crack propagation effects, and flaw interactions tend to change the value of the Weibull modulus. Indeed, Stienstra⁸ has calculated that certain crack propagation effects tend to increase the value of the measured Weibull modulus.

The idea that certain aspects of the microstructural failure modes may be inferred from the value of the Weibull modulus is demonstrated to a preliminary level of certainty. That further aspects of microstructural failure modes could be inferred from the Weibull failure parameters through material microstructural failure simulation and observation is of great interest in terms of fabrication process improvement and understanding of failure strengths in brittle materials.

Experimental Data Reduction for Weibull Parameters

Four-point bend bar tests were conducted on specimens fabricated from beryllium oxide (BeO) components to measure the effect of specimen surface finishing. One set of specimens had a machine ground surface, while the second set were polished to a mirror finish. The specimens were all nominally size A according to the ASTM specification C-1161.⁹ Both sets of specimens were fabricated and tested in accordance with this specification, and the data reduction followed the guidelines of ASTM C-1239.¹⁰ The size A specimens are nominally 2.0 mm wide, 1.5 mm deep, and 25 mm long. In the four-point bend test apparatus the inner span is 10 mm, with the outer span twice that. A good discussion of the issues of accuracy and reliability in the testing aspects for ceramic materials is found elsewhere.¹¹

A third set of specimens is referred to in the results as the BeO annealing study. Annealing temperatures were applied to the material to see if there could be positive results from pore closure, crack healing, and residual stress relief due to material flow at elevated temperatures. If there were such effects, they are subtle, and these data points are included for comparison purposes.

Figure 1 shows the four-point bend tests results with failure stress in pounds per square inch as a function of probability of failure. There was an

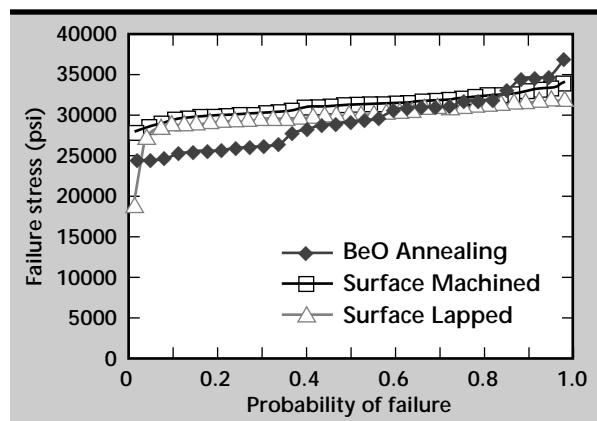


Figure 1. BeO four-point bend specimen failure comparisons.

additional surface-machined specimen which broke in fabrication, with the remaining 30 points plotted. The CARES program¹ and the 'WeibPar' program from CRT associates¹² were used for the data reduction. The two programs agree to within numerical error.

The BeO failure data for the polished or lapped samples, along with the plots of the curve fit parameter from the WeibPar program are shown in **Fig. 2**. Using the maximum likelihood estimators as the curve fit technique, the Weibull modulus, m , is calculated to be 25.83, with the Weibull scale parameter 22,070 psi*(in³)^(1/25.83). The single low data point is an obvious 'outlier' from the distribution. It lies far from the other data points and from the curve fit parameters. When a linear regression curve fit is used to better capture this outlying point, m is calculated to be 10.77 while the scale parameter is 15,770 psi*(in³)^(1/25.83).

The large change in material parameters with the change in curve fit techniques is understandable, but also somewhat disturbing. It is understandable in that the linear regression curve fit technique biases the curve fit to the low-level points. It is disturbing in that it reduces the confidence level that the curve fit parameters represent real material properties.

From the curve fits, and the persistence in each case of low stress level failures, it appears that

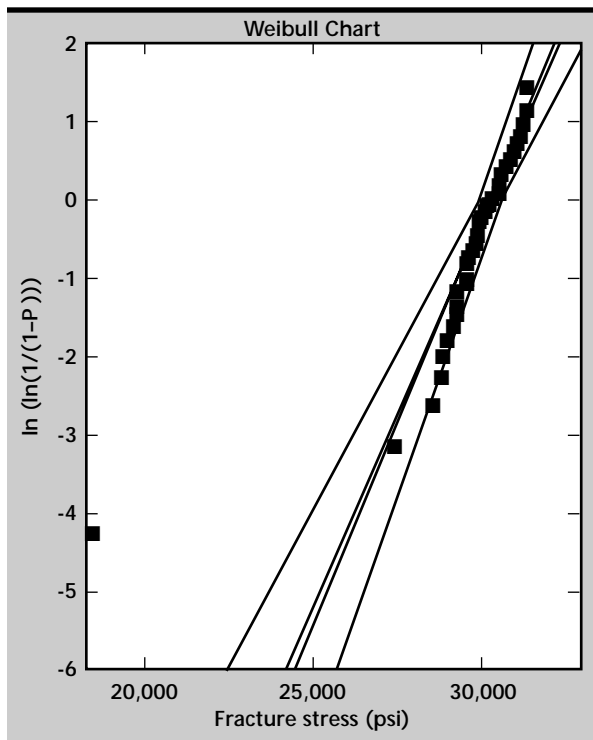


Figure 2. Weibull Chart for lapped BeO specimens.

there is a second type or distribution of flaws which is less prevalent but whose consequence on failure is much more severe. The existence of this second type of flaw distribution must be confirmed by additional tests on larger samples to bolster predictions of its effects. This represents a real difficulty in applying Weibull statistics to ceramic materials. It also offers the hope that if the larger defects could be eliminated through better material processing, very significant improvements could be made in raising the expected design stress of BeO components.

Fractography of the Failed BeO Specimens

Fractography of the failed samples is required to verify the assumption that the flaws that caused failure were distributed throughout the volume of the material, and not just present on the surface. The Military Handbook 790¹³ provides useful guidelines and information regarding the characterization of failed ceramic surfaces.

The surface of the as-machined BeO specimen is seen in **Fig. 3** as a micrograph from a scanning electron microscope. The areas marked (a), (b), and (c) show the features of machining striations, a machining induced crack, and machining debris on the surface.

The as-lapped specimens show polished smooth surfaces mixed with voids. The voids are of various sizes and shapes as shown in **Fig. 4a** and their internal structures reveal the individual grains or clusters of grains as shown in **Fig. 4b**. **Figure 4b** is the enlarged view of the rectangular area marked in **Fig. 4a**. Evidences of unbonded boundaries between two neighboring grains are shown by arrows in **Fig. 4a** and unbonded three-grain junctions are shown in **Fig. 4b**.

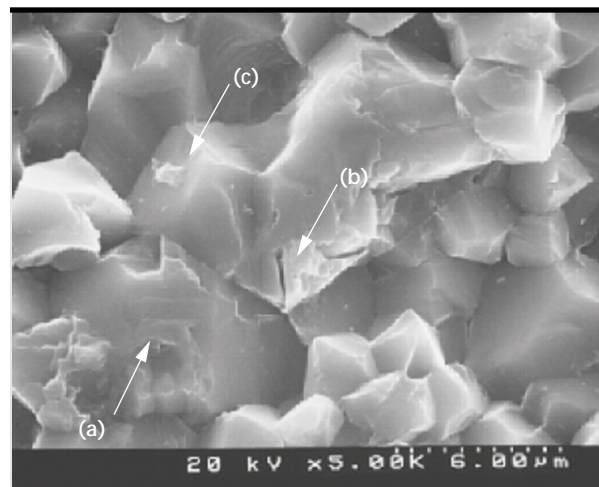


Figure 3. Surface of the as-machined BeO samples. Areas marked (a), (b), and (c) show features of machining mark, machining-induced crack, and machining debris, respectively, on the machined surface.

From **Figs. 3 and 4**, the average grain size of the present material was determined by a simple line intercept method to about 3 μm . It is not clear whether the microstructure shown in **Fig. 4a** represents the true bulk microstructure or near-surface microstructure where the machining damage is still visible. If the voids seen on the surface in **Fig. 4a** are a part of the true bulk structure, then the material must contain a fairly high content of voids. If the voids are due to the machining damage and left in place by insufficient lapping, a further lapping is necessary to remove the machining-induced damage and reach the true bulk microstructure. The volume fraction of the voids in **Fig. 4a** is calculated to be 14% by the line intercept method.

For purposes of the data reduction it is assumed that the specimen critical flaws are volumetrically distributed.

Figure 5 shows the fracture path in the as-lapped sample. The arrows indicate the propagation

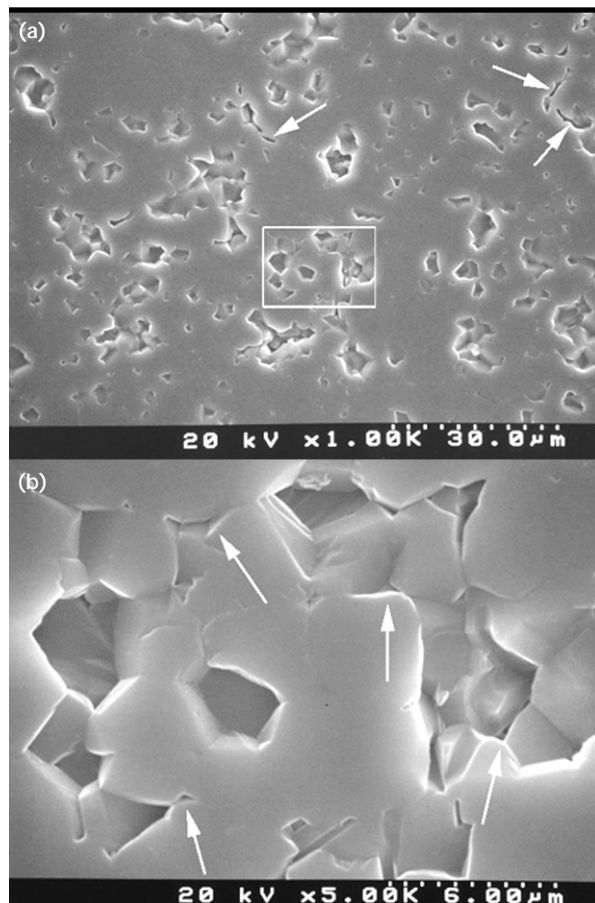


Figure 4. (a) Microstructure of the as-lapped surface. Arrows indicate sharp crack-like unbonded boundaries between two neighboring grains, where a fracture can be initiated. (b) Enlarged view of the area marked by a rectangle in Fig. 4a. Arrows indicate where three or more grains meet and form an unbonded junction, which can act as a fracture initiation point.

of a secondary crack as revealed on the “tensile” surface of the four-point bend test specimen. The fracture path is seen to follow the grain boundaries and connect the voids. From the micrographs of the failure surface it seems apparent that the failure of the ceramic material on the microstructural level is strongly influenced by the fracture toughness of the grain boundaries, as compared to the matrix of the crystalline materials, and that voids play a large role in both crack initiation and crack propagation.

Implementation of the Weibull Failure Predictions in NIKE2D

The implementation of the Weibull distribution for uniaxial stress is quite straightforward for the two-parameter failure model. For arbitrary 3-D stresses an equivalent uniaxial stress is calculated, and the analysis becomes complicated by additional choices. Two methods for arriving at an equivalent uniaxial stress, given a 3-D stress field, were examined in depth and implemented in NIKE2D. These methods are termed the principle of independent action, and the Weibull averaged normal tensile stress.

For the principle of independent action, if the principal stresses are tensile, this approach gives a probability of failure as

$$P_{fv} = 1 - \exp \left[-k_{wv} \int_V \left(\sigma_1^m + \sigma_2^m + \sigma_3^m \right) dV \right]$$

where m is the Weibull modulus and k_{wv} is $(\sigma_{oV})^{-m}$.

The Weibull normal stress method gives more conservative estimates of the probability of failure.¹ The risk of fracture at any point in the specimen is a function of the tensile normal stress distribution, σ_n ,

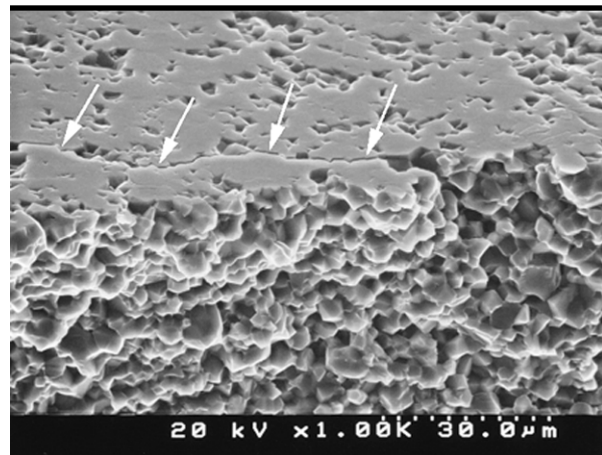


Figure 5. Fracture path in the as-lapped sample. Arrows indicate the propagation of a secondary crack as revealed on the “tensile” surface of the four-point bend test sample. The fracture path is seen to follow the grain boundaries and connect the voids.

on all planes passing through the point.¹⁴ All the planes passing through the point can be represented, in turn, as normal vectors of unit length, which terminate on a sphere of unit radius in principal stress space. For an arbitrary multiaxial load on an element of material volume, the principal stresses acting on the element must be projected onto those portions of the unit sphere for which the averaged stress is tensile, under the assumption that an average compressive stress is not harmful to the material.

This projected average quantity, σ_n , is equal to:

$$\sigma_n = (\cos\phi)^2 [\sigma_1 (\cos\psi)^2 + \sigma_2 (\sin\psi)^2] + \sigma_3 (\sin\phi)^2$$

where σ_1 , σ_2 , and σ_3 are the principal stresses, ϕ is the azimuthal angle, and ψ the polar angle for the unit sphere, as seen in **Fig. 6**. The equivalent uniaxial stress is then given by

$$\overline{\sigma_n^m} = \frac{\int_A \sigma_n^m dA}{\int_A dA}$$

where dA is an elemental area on the reference sphere of unit radius.

The failure probability, P_F , is then given by:

$$P_{IV} = 1 - \exp \left[- \int_V k_{wpV} \overline{\sigma_n^m} dV \right]$$

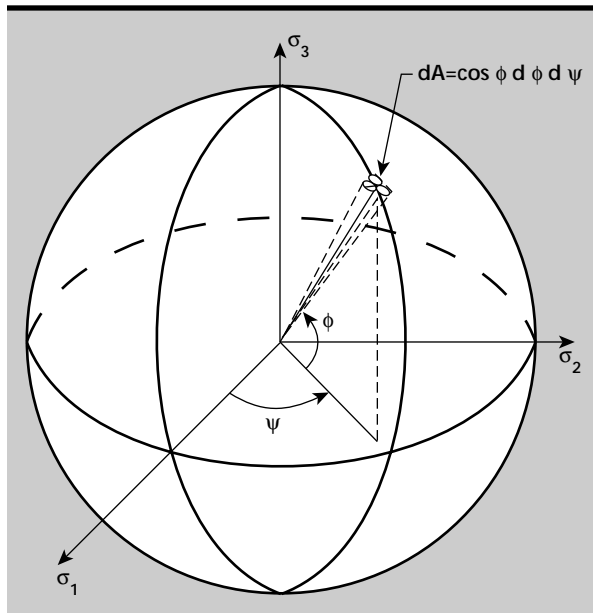


Figure 6. Angle definition for calculation of the Weibull averaged normal stress.

where dV is a differential volume in the specimen over which the principal stresses are approximately constant. In the preceding equation, m is the Weibull modulus as before, and k_{wpV} is the polyaxial Weibull crack density coefficient. The relationship between k_{wV} and k_{wpV} is found by equating the failure probability for uniaxial loading to that obtained for the polyaxial stress state when the latter is reduced to a uniaxial condition.¹ The result is

$$k_{wpV} = (2m+1) k_{wV}.$$

These integral equations can be replaced with a discrete expression suitable for implementation in a numerical algorithm on a computer:


$$\int_0^{\pi/2} \int_{-\pi/2}^{\pi/2} (\sigma_n)^m \cos\phi \, d\phi \, d\psi = \sum \sum (\sigma_n)^m \cos\phi \, \Delta\phi \, \Delta\psi$$

where the functions $\cos\phi$ and σ_n are evaluated at $\phi = -\pi/2 + (j-1)\Delta\phi$ and $\psi = (i-1)\Delta\psi$.

These expressions have been incorporated into a version of NIKE2D. The results of these efforts are currently being verified.

References

1. N. N. Nemeth, J. M. Mandersheid, and J. P. Gyekenyesi (1990), "Ceramic analysis and reliability evaluation of structures (CARES) users and programmers manual," NASA TP-2916, National Aeronautics and Space Administration.
2. W. Weibull (1951), "A statistical distribution function of wide applicability," *ASME J. of Appl. Mech.*, September, pp. 293-297.
3. E. Y. Robinson (1964), "Some theoretical and experimental aspects of design with brittle material," Lawrence Livermore National Laboratory, Livermore, California (UCRL-7729), August 3.
4. P. Kittl and G. Diaz (1988), "Weibull's fracture statistics, or probabilistic strength of materials: state of the art," *Res Mechanica* **24**, pp. 99-207.
5. T. L. Anderson (1995), "Fracture mechanics: fundamentals and applications," 2nd Edition, CRC Press, p. 292.
6. R. O. Ritchie, J. F. Knott, and J. R. Rice (1973), "On the relationship between critical tensile stress and fracture toughness in mild steel," *J. of the Mechanics and Physics of Solids*, Vol. **21**, pp. 395-410.
7. J. W. Hutchinson (1968), "Singular behavior at the end of a tensile crack tip in a hardening material," *J. of the Mechanics and Physics of Solids*, Vol. **16**, p. 13.

8. Auen-Stienstra, D. I. (1990), "Stochastic micromechanical modeling of cleavage fracture in the ductile-brittle transition region," Texas A&M Dissertation, August, p. 99.
9. American Society for Testing and Materials, "ASTM C-1161-94, Standard Test Method for Flexural Strength of Advanced Ceramics at Ambient Temperature," 1994, Philadelphia, Pennsylvania.
10. American Society for Testing and Materials, "ASTM C-1239-94a, Standard Practice for Reporting Uniaxial Strength Data and Estimating Weibull Distribution Parameters for Advanced Ceramics," 1994, Philadelphia, Pennsylvania.
11. D. C. Cranmer and D. W. Richerson (1998), "Mechanical testing methodology for ceramic design and reliability," Marcel Dekker, New York, New York.
12. S. D. Duffy and E. H. Baker (1997), "Weibull parameter estimation: theory and background information," CRT Technologies, November.
13. Military Handbook, MIL-HDBK-790, March 1993, Information Handling Services, DODSTD Issue DW9805.
14. O. Vardar and I. Finnie (1975), "An analysis of the Brazilian disk fracture test using the Weibull probabilistic treatment of brittle strength," *Int. J. of Fracture*, Vol. **11**, No. 3, June, pp. 495-508. 



Lattice Boltzmann Modeling of Spherical Solutes in Confined Domains

David S. Clague
Electronics Engineering Technologies Division
Electronics Engineering

The primary aim of this effort is to develop a unique lattice Boltzmann (LB) simulation capability to understand the dynamic behavior of macromolecules, cells, and spores in bounded flow domains. The proposed effort for FY-99 included benchmark studies with spherical solutes to verify the utility and accuracy of the LB method. These studies include calculating the hydrodynamic force and torque acting on a stationary, spherical particle in bounded flows and calculation of the hydrodynamic force acting on a mobile particle in the presence of a stationary wall. This year's results show that the LB accurately solves these non-trivial flow problems. Selected results for stationary and mobile macroparticles are presented. An additional milestone that was achieved ahead of schedule is that the 2- and 3-D LB codes with macroparticles are now parallel.

Introduction

Many fluid-based processes used, for example, in the chemical and pharmaceutical industries, involve the transport of nanometer-to-micron sized particles in confined flow domains. To augment the design process and to better understand the fundamental behavior of particles in such systems, one needs to be able to model and simulate the flow field, the mechanisms for transport phenomena associated with charged, polar particles in the presence of electric fields, and the other relevant force interactions. In the work presented here, the LB method was chosen to solve these complex problems.

As opposed to more traditional continuum methods, such as Boundary Element and Finite Element methods, the LB method is a mesoscopic approach. Specifically, when using this approach, one keeps track of the single-particle velocity distribution function on a regular mesh of lattice sites. This approach derives from the Boltzmann transport equation, which is the conservation equation found in the kinetic theory of gases.¹ Under the appropriate flow conditions, for example, small Mach and Knudsen number flows, the LB method accurately captures the hydrodynamic behavior of viscous fluids.^{2,3} Additionally, because the approach is discrete, it can easily handle bounding walls, mobile particles, external fields, and finite Reynolds numbers.

Another convenient feature of the LB method is that fluid velocity disturbances are transmitted locally from lattice site to lattice site. As a consequence, the *fluid phase* is easily parallelizable relative to the conventional methods such as finite element methods.

Progress

The selected results presented here include calculating the hydrodynamic force acting on an infinite medium of equispaced stationary spheres, a stationary sphere in a bi-periodic simulation cell with bounding walls at the top and bottom of the flow cell, and a single sphere translating in direct approach toward a stationary planar surface. The results presented were calculated using a parallel LB code developed during FY-99. The parallelization includes both the fluid phase and the particulate phase. All simulations were performed on multiprocessor machines using MPI. Simulations were conducted on local Dec Alpha clusters and on an IBM SP2 machine with multiple processors.

The benchmark studies on stationary or captured macroparticles produced interesting results. Specifically, when passing a flow through an infinite medium of stationary periodic spheres, the force on the test sphere increases with increasing solids fraction. **Fig. 1** shows the unit cell that gets replicated throughout space forming an infinite medium of equally spaced spheres.

As shown, the hydrodynamic drag force calculated using the LB method is nearly identical with the well-accepted result of Zick and Homesy.⁴ Also, it is important to note that the drag force increases with increasing solids fraction. In contrast, when placing the sphere in a flow channel bounded at the top and bottom with parallel walls, the drag force decreases as the solids fraction is increased. **Figure 2** shows a sphere of radius a bounded by parallel walls separated by distance H . Also shown are the upstream and downstream periodic images of the central sphere.

The result shown in **Fig. 2** shows the opposite trend in drag force as was seen in **Fig. 1**. For example, the hydrodynamic drag force decreases as the spheres are brought closer together, or for increasing solids fraction. As shown, it is necessary to separate the periodic spheres by a minimum of 30 sphere radii to ensure that the central sphere is hydrodynamically de-coupled from its neighbors. The flow system depicted in **Fig. 2** is very much like particle capture apparatus used in many flow systems that contain particulates; hence, the result presented here can be

used to calculate the hydrodynamic force acting on capture macroparticles in an actual flow cell.

Essentially, the observed difference between the drag force shown in **Figs. 1** and **2** has two explanations. First, for the flow configuration in **Fig. 1**, the fluid flux area between periodic spheres decreases with increasing solids fraction which in turn increases the drag force for a constant pressure gradient. In contrast, the fluid flux area for the flow configuration in **Fig. 2** is constant for all solids fractions; hence, there is no change in the bulk fluid velocity with change in solids fraction. The second reason that the drag force in **Fig. 2** decreases with increasing solids fraction is a form of “drafting.” More specifically, the fluid velocity disturbance caused by the upstream sphere does not recover to the bulk fluid velocity before it encounters the next neighboring downstream sphere. Others have observed this same “drafting-like” phenomenon in sedimentation problems at low Reynolds numbers.

The final result presented here is for a sphere translated at constant velocity in direct approach to a stationary wall (**Fig. 3**).

Figure 1.
Hydrodynamic drag force acting on a periodic array of spheres. (a) Flow configuration. (b) Comparison of drag force calculated using the LB method with the well-accepted result of Zick and Homesy.

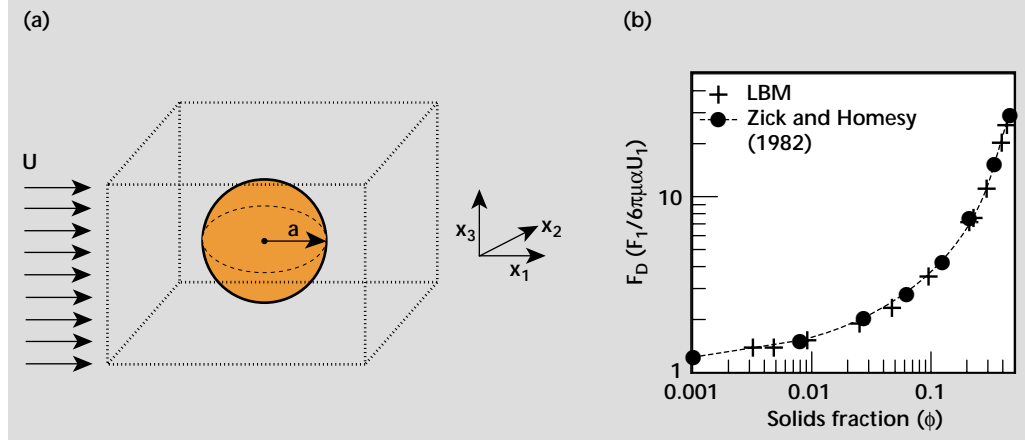
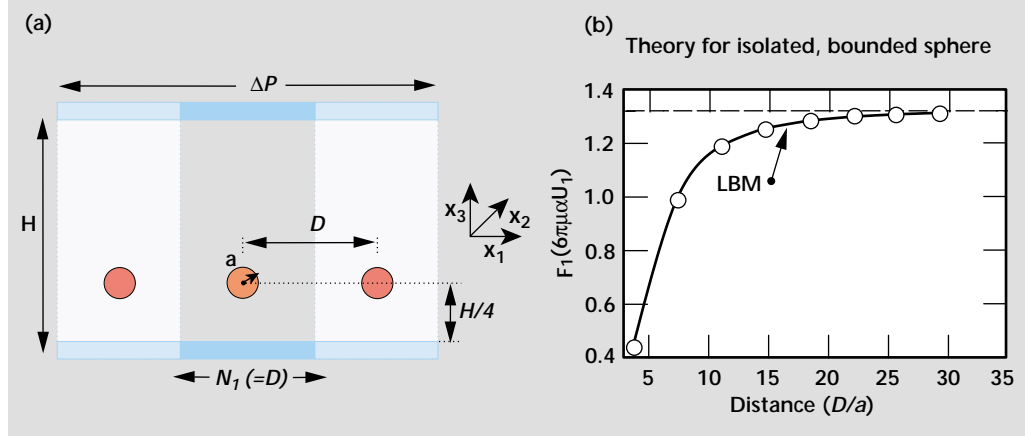
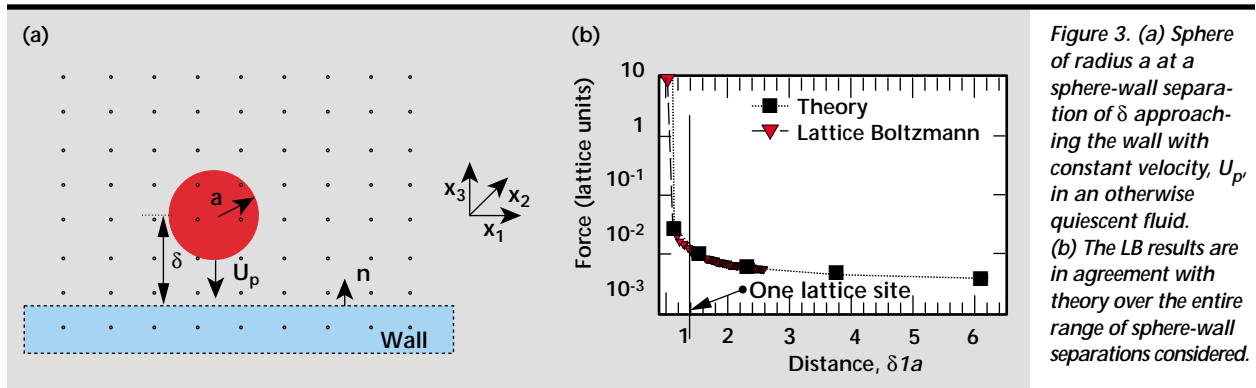


Figure 2.
(a) Hydrodynamic force acting on a stationary spherical body in Poiseuille flow. (b) Hydrodynamic drag force acting on the central sphere as a function of the number of sphere radii between the central sphere and the nearest periodic neighbor.





The LB results are in excellent agreement with theory even at infinitesimal sphere-wall separations. This demonstrates that the LB approach accurately captures hydrodynamic interactions for mobile macroparticles.

In addition to the selected results presented above, the LB simulation capability has also been used successfully to study fluid behavior for flow through conduits with irregular cross-sectional geometry. Furthermore, simulations are in progress to study the behavior of a single macroparticle in Poiseuille flow in bounded conduits. Work is also in progress to include external fields such as gravitational and electromagnetic fields in the parallel LB codes.

Future Work

The LB simulation capability has developed into a useful predictive tool. The work for this coming year will include validation of the LB method with experimental results and inclusion of new physics relevant to laboratory needs. The validation studies

will include the study of suspension behavior in rectangular conduits and particle capture due to external fields.

In FY-00, intermolecular force will be included in the LB simulation capability and the resulting suspension behavior will be studied.

References

1. Landau and Lifshitz (1989), "A course of theoretical physics, Volume 10," Pergamon Press.
2. D. S. Clague, *et al.* "On the hydraulic permeability of (un) bounded fibrous media using the lattice Boltzmann method," submitted to *Phys. Rev. E*.
3. D. S. Clague (1998), "Lattice Boltzmann simulation of complex fluids," *Engineering Research, Development and Technology*, Lawrence Livermore National Laboratory, Livermore, California (UCRL-53868-98).
4. A. A. Zick and G. M. Homesy (1982), "Stokes flow through periodic arrays of spheres," *J. Fluid Mech.* **115**, 13.



Modeling of Anisotropic Inelastic Behavior

Daniel J. Nikkel, Jr. and Deepak S. Nath
*New Technologies Engineering Division
 Mechanical Engineering*

Arthur A. Brown and James Casey
*University of California at Berkeley
 Berkeley, California*

An experimental capability, developed at Lawrence Livermore National Laboratory (LLNL), is being used to study the yield behavior of elastic-plastic materials. The objective of our research is to develop better constitutive equations for polycrystalline metals. We are experimentally determining the multidimensional yield surface of the material, both in its initial state and as it evolves during large inelastic deformations. These experiments provide a more complete picture of material behavior than can be obtained from traditional uniaxial tests. Experimental results show that actual material response can differ significantly from that predicted by simple idealized models. These results are being used to develop improved constitutive models of anisotropic plasticity for use in continuum computer codes.

Introduction

At a microstructural level, polycrystalline metals are composed of aggregates of individual crystals, each of which has its own orientation and properties. When subjected to loading, metals initially exhibit reversible deformation, due to the stretching of the lattice. When the loads become sufficiently large, permanent deformations can occur through a number of mechanisms, such as dislocation motion, twinning, or grain boundary sliding. As a consequence of having randomly distributed grain orientations, annealed polycrystalline metals typically exhibit isotropic behavior with respect to a reference configuration; that is, at a given point in the material, the material response of a specimen in any direction is the same. This includes the elastic behavior and the initial yield behavior. However, significant processing of materials, or even moderate plastic deformations, can cause grains which were initially randomly oriented to become aligned, resulting in behavior which is anisotropic, where material response in different directions is quite different.

The ability of numerical simulations to predict the behavior of systems involving materials undergoing large deformations is contingent upon having

a realistic model of the material behavior. Such models must be accurate in the full range of possible loading conditions to which the materials may be subjected. Use of overly simplified models in regimes where they are not well suited can seriously compromise the validity of a simulation. Many problems of engineering interest involve metals undergoing large deformation under multiaxial states of stress and the need for reliable models for these applications can hardly be overemphasized. Experimental data demonstrate that simple models for plasticity commonly used in numerical codes do not accurately predict material behavior under these conditions.

Engineering models of polycrystalline metals generally omit microstructural details and describe the effective macroscopic behavior in terms of a phenomenological continuum model. Viewed from the macroscopic perspective, the initial material response is path-independent and there is a one-to-one correspondence between stress and strain. However, if the deformation or loads become sufficiently large, the material begins to exhibit plastic behavior. There is no longer a one-to-one correspondence between stress and strain, the response is dependent on the loading path taken to reach a given state of deformation, and permanent deformation

remains after external loads are removed. This gives rise to the theoretical idealization of an elastic-plastic material, and in particular, to the notion of a yield function¹ denoted by

$$f(s_{kl}, e_{kl}^p, \kappa, \dots) = g(e_{kl}, e_{kl}^p, \kappa, \dots) \quad (1)$$

Here, s_{kl} denotes the components of the stress tensor; e_{kl} denotes the components of the strain tensor; e_{kl}^p denotes the components of the plastic strain tensor; κ is a scalar measure of work hardening; and the ellipses represent other inelastic state variables which may be present, depending on the complexity of the chosen constitutive theory.

The yield function is a key ingredient of the constitutive theory of elastic-plastic materials. The condition $f = 0$ (or $g = 0$) describes the boundary between stresses (or strains) that result in only elastic behavior, and those which result in inelastic behavior. For fixed values of the inelastic variables, the yield condition can be interpreted geometrically from the point of view of stress space (or strain space), as a surface which bounds the region in which only elastic behavior occurs (the elastic region). When the loading path intersects the yield surface and tries to cross it, inelastic behavior occurs and plastic deformation results. The current state never moves outside the yield surface, but instead the surface is carried along with it. Typically, the yield surface changes both in shape and size as the inelastic deformation increases. Measured yield surfaces for three of the states along the complex stress path depicted in **Fig. 1a** are shown in **Fig. 1b**.

In addition to the yield function, the constitutive theory includes evolution equations for the inelastic variables during loading ($\dot{g} = 0$, $\dot{g} > 0$). Thus, for the plastic strain, we have

$$\dot{e}_{kl}^p = \rho_{kl}(e_{mn}, e_{mn}^p, \kappa, \dots) \hat{g}, \quad \hat{g} \equiv \frac{\partial g}{\partial e_{mn}} \dot{e}_{mn} \quad (2)$$

Here ρ_{kl} is a constitutive response function that is independent of the rates of stress or strain. For a broad class of materials, under a physically reasonable assumption regarding work in closed cycles in strain space, ρ_{kl} can be replaced with the product of a scalar function and the normal to the yield surface in stress space, thus requiring the specification of only one additional scalar response function.¹ For special classes of materials, this scalar function is determined from the yield function and hardening and does not require an independent specification.

Most plasticity models implemented into numerical codes for metals use a yield criterion which corresponds to a fixed shape of the yield surface (for example, elliptical in the case of the Mises yield

criterion). What distinguishes different models is how the yield surface evolves (for example, it may translate rigidly, or alternatively change its size while maintaining its shape, or follow some combination of these simple hardening laws). Many simulations are run with a model that assumes an elliptic yield surface of fixed aspect ratio that only changes in size due to hardening. While the initial yield surface of isotropic materials may be represented reasonably well by an ellipse, subsequent to even moderate plastic deformation, the shape of the yield surface in real materials can change significantly (**Fig. 1b**). For this reason, simple representations of the yield function will be satisfactory only under very restrictive loading conditions (such as monotonic or uniaxial), and are inadequate for general multiaxial loading conditions, especially

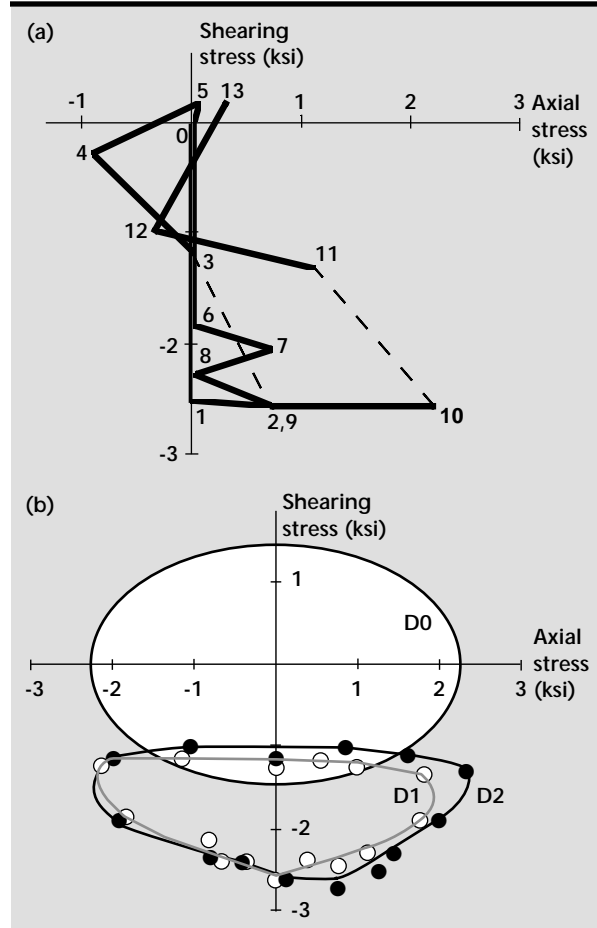


Figure 1. (a) Imposed sequence of loading states for yield surface measurement on a single specimen, identified as "D" in Table 1. (b) Measured points on yield surfaces in 2-D stress space from a single 1100 aluminum specimen subjected to the loading states in Fig. 1a. For clarity, only the surfaces measured at load states 0, 1, and 2 in Fig. 1a have been shown. The subsequent yield surfaces show significant deviation from an idealized ellipse even though the strains involved are moderately small.

when loads can reverse and change direction during the history of loading.

In view of these considerations, and motivated by the fact that the vast majority of available experimental data on polycrystalline metals are for uniaxial (and generally monotonic) loading, our group developed an experimental capability to map out the yield surface at various fixed states of large inelastic deformation under multiaxial loading. By determining the yield surfaces on a single specimen at multiple fixed states, the evolution of the yield surface during plastic deformation can be observed. These data provide the basis for developing improved constitutive equations for phenomenological descriptions of polycrystalline metals.

Progress

This project is a combination of a program of novel experiments characterizing inelastic material behavior together with an effort to develop better material models for implementation into numerical analysis codes. The primary approach has been to obtain improved experimental data for the macroscopic response as a guide to the development of better phenomenological models. During FY-99, we have further refined the experimental procedures developed in previous years, and have obtained some important new data.

In addition to this direct macroscopic approach, we have begun to examine the alternative strategy of incorporating information from lower length scales where explicit consideration is taken of material microstructure, and we have also begun to explore how this mesoscale description can be homogenized to obtain improved macroscopic models.

Further, work on general issues related to numerical implementation of anisotropic plasticity models has been pursued, in the context of both purely Lagrangian and arbitrary Lagrangian-Eulerian (ALE) formulations.

Experiments

The experimental portion of this project involves determination of the yield surface under multiaxial states of loading, using thin-walled tension-torsion specimens with a 2-in. inside diameter. The experimental determination of the yield surface of the material is carried out by loading a specimen under multiaxial conditions and probing until the point of yield is reached, then backing off and probing in a different direction in stress space (and in strain space) until the next yield point is found. This process is repeated until the entire surface is

mapped out. The sensitive nature of the measurements being made requires careful attention to the issues of specimen design and preparation, experimental methodology, and interpretation of the data. The general description of the experiments and the difficulty in carrying out these measurements have been discussed previously,^{2,3} and this year a comprehensive description of the experimental procedure has been compiled.⁴

Ideally, a measured yield surface represents the boundary between elastic and inelastic behavior at a given elastic-plastic state. As a practical matter, the identification of the yield point requires loading somewhat beyond the elastic region so that the inelastic behavior becomes evident. The procedure which has been developed can detect yield without producing a plastic strain much greater than 5×10^{-6} (5 μ -strain), and yield surfaces can be obtained for multiple elastic-plastic states from a single specimen. It is clear from the data that the yield surfaces determined from multiaxial loading tests are strongly dependent on the method used to identify the yield point. Many different definitions of the yield point are possible; these vary in both experimental complexity and in the amount of plastic deformation that is induced during the determination of yield at a given point. The measured yield surfaces can vary from appearing roughly isotropic when a coarse large off-set or back-extrapolation method is used, to clearly exhibiting material anisotropy when a small off-set definition of yield is used (**Fig. 1b**). The definition of yield point is related to the idealized way in which the theoretical model represents real material behavior. These issues are discussed more fully elsewhere.⁴ After investigating various alternatives, we adopted a 5 μ -strain offset definition of yield as the most meaningful for our current interests.

In addition to generating data from the measurement of yield surfaces, we are also exploring the fundamental question as to the proper definition of plastic strain in the context of large inelastic deformations.^{5,6} When yield surfaces have moved so that they no longer enclose the origin in stress space (for example, yield surfaces D1 and D2 in **Fig. 1b**), the material cannot be unloaded to zero stress without causing new plastic deformations.

The traditional way of defining plastic strain is to identify it with the residual strain remaining when the load is removed. This definition arose intuitively from consideration of uniaxial tests with small deformation, but it is clearly inadequate in the situation of more general states of loading where the yield surface no longer encloses the origin in stress space. Plastic strain is not among the set of kinematic

variables that come from classical continuum mechanics. Since it is a primitive variable in the constitutive theory, one must be able to identify it unambiguously for the theory to be meaningfully predictive and not simply a sophisticated curve-fit.

Previously, an experimental methodology was proposed for verifying the validity of our prescription for identifying plastic strain, in cases where the origin lies outside the stress space yield surface. It is defined as the point in strain space corresponding to the point on the yield surface in stress space closest to the origin.^{5,6} This method exploited the property from **Eq. 2** that the direction of the increment in plastic strain is independent of the size and direction of an applied increment in stress or strain. By comparing the directions of the plastic strain increment for two different loading directions originating from the same state, the prescription can be verified. The high precision necessary to be able to make the measurements required for this verification was a driving force governing the refinement of experimental procedures. Three different verifications (requiring the determination of six distinct yield surfaces) have been completed which do in fact validate this prescription for identifying plastic strain.

Characterization of Experimental Results

One consistent observation in the experimental data with a 5 μ -strain off-set definition of yield (such as in **Fig. 1b**) is a distortion of the shape of the yield surface after moderate plastic deformation when loading away from the origin. The initial yield surface is close to elliptical in shape. For subsequent yield surfaces, the side away from the loading point (and closest to the origin) tends to flatten, while it tends to elongate near the loading point, resulting in a D-shaped surface. There are classical closed curves, such as the folium, the piriform, and the pear-shaped quadric, that are somewhat D-shaped.⁷ However, after some effort it was found not possible to use these functions to represent the data in a robust manner. An alternative approach to using a single analytical function to represent the yield surface in two dimensions is to approximate it with a collection of smooth intersecting segments. As a curve-fitting scheme this approach is capable of matching a given set of data with any desired degree of accuracy, but such a representation provides little physical insight towards the development of improved models. Moreover, describing yield surface evolution in a general way becomes problematic, and it also presents difficulties in terms of numerical implementation.

To characterize the data, and to give some direction to the development of improved models, “best fits” to the data were performed with polynomial functions representing closed curves. Even for quadratic polynomials this is a nontrivial exercise. In the usual routine task of fitting a function to discrete x - y data, an error function is defined and the unknown coefficients are determined by minimizing the error. The resulting system of equations is linear and a unique solution is guaranteed.

For the problem at hand, however, the data pairs (x_i, y_i) cannot be approximated by an explicit function of the form $y = f(x)$; rather they must be represented by implicit functions of the form $f(x,y) = 0$. To fit an implicit function to the data, an error function is defined as usual, but the system of equations that must be solved to determine the unknown coefficients is highly nonlinear and there is no guarantee of uniqueness as in the linear case. Solving the system of nonlinear equations numerically is a complex undertaking requiring sophisticated solution techniques.

Quadratic fits to all the data sets were obtained by using a package for orthogonal distance regression, ODRPACK, which is publicly available from NIST⁸ (**Fig. 2**). The program finds the parameters that minimize the sum of the squared weighted orthogonal distances from a set of observations to the curve or surface determined by the parameters. This procedure provided a method of characterizing the data in terms of change in position, size, and orientation of subsequent yield surfaces. Results of the data reduction for various measured yield surfaces are shown in **Table 1**. A second-order polynomial is sufficient for gauging the size, location, and some sense of the “orientation” of the yield surface, but it cannot adequately represent features such as relative elongation and flattening of the yield surface. Higher order

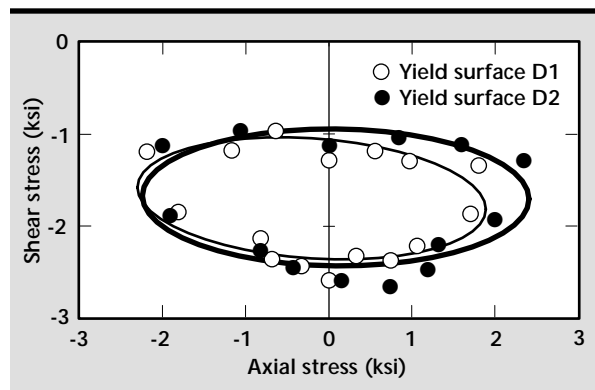


Figure 2. Second-order polynomial fits to yield surface data. Quadratic functions can capture the trends in size, location and some sense of orientation, but provide no characterization of the distortion into the observed D-shapes.

function fits could likely provide better representations of the D-shaped data curves, but ODRPACK was unable to give reliable fits in a unique, robust manner.

Numerical Modeling

The Mises yield condition, which is a quadratic polynomial in the deviatoric stress components, is known to agree well with experimental data on annealed (isotropic) polycrystalline metals, and it also has a physically appealing interpretation in terms of distortion energy. For anisotropic materials, the most general quadratic yield function representing a smooth initial yield surface which reduces to the Mises yield function in the special case of isotropic materials is of the form⁹

$$f = B_{klmn} S_{kl} S_{mn} - \kappa^2, \quad (3)$$

where due to symmetries the coefficients B_{klmn} have 21 independent components. If, as is commonly done, the further assumption is made that the yield behavior is independent of the mean stress (pressure), then the stress tensor in **Eq. 3** can be replaced by its deviatoric part and the coefficient tensor can be replaced by a reduced tensor which has 15 independent coefficients. As mentioned previously, a quadratic representation of the yield surface can reasonably capture its location, size, and a sense of the orientation observed in real materials, but it does not capture the distortion of the yield surface from elliptical to D-shaped. Nevertheless, **Eq. 3** is much more general than the Mises yield function, and it contains as special

cases other anisotropic yield models which have been implemented into LLNL codes.

An anisotropic plasticity model with the yield function in **Eq. 3** was implemented into the parallel version of the code ALE3D. **Figure 3** shows a Lagrangian numerical simulation of a Taylor impact test using an anisotropic tantalum cylindrical projectile. The predicted ovaling of the impact footprint is in agreement with experimental data.¹⁰

Since a basic feature of anisotropic behavior is that the response depends on the material direction, it is necessary to track material directions during deformation. While this is a straightforward issue within a Lagrangian formulation, implementation of anisotropic plasticity models in the context of an ALE formulation is complicated by the fact that nodes of a mesh are not material points, and the element edges are not material curves. An alternative approach must be used to keep track of material directions. Material direction vectors m_i can be stored as element-based variables, and can then be updated during the Lagrangian step according to

$$\dot{m}_i = (L_{ij} - (L_{kl} m_k m_l) \delta_{ij}) m_j, \quad (4)$$

where L_{ij} are the components of the velocity gradient tensor. These element-based quantities can then be updated as other history variables during the advection step. There are increased memory storage costs associated with modeling anisotropic plasticity. Along with storing the material direction vectors and any additional constitutive variables, properly invariant formulations of anisotropic

Table 1. Characteristics of yield surface data obtained from quadratic fits. The yield surface designation comes from a specimen identification letter and a state number.

Yield surface	Offset Def. (μ -strain)	e_{11}^p (μ -strain)	e_{12}^p (μ -strain)	Center (psi)	Orientation angle ($^\circ$)	Subaxis 1 (psi)	Subaxis 2 (psi)	Aspect ratio
A1	15	300	0	(857, -57)	29.8	1250	1460	0.86
A2	15	8690	7128	(871, 947)	-16.3	3000	1380	2.17
B0	1	0	0	(-87, 94)	3.91	1770	1060	1.67
B1	1	1	-17	(-11, -263)	2.23	1840	890	2.07
B2	1	92	-2055	(37, -1809)	-1.12	1550	370	4.19
C1	1	-4578	102	(-3425, 114)	24.7	770	1170	0.66
C2	1	-5008	234	(-3713, 82.4)	40.4	640	1050	0.61
D1	5	5	-2020	(-209, -1698)	-7.2	2230	650	3.43
D2	5	30	-2100	(80, -1698)	-0.9	2330	750	3.11
D4	5	30	-2010	(84, -1230)	-3.3	2610	770	3.39
D10	5	1030	-3660	(1250, -1918)	14.0	2380	670	3.55
D12	5	990	-3560	(1145, -1459)	26.0	2570	1070	2.40

models also require storing additional kinematical quantities such as the rotation and stretch tensors.

Alternative Approaches Using Mesoscale Homogenization

The approach described earlier seeks to develop a macroscopic phenomenological model directly from measurements of the macroscopic inelastic behavior. An alternative strategy is to examine the governing processes in terms of microphysical behavior at lower length scales, and then to deduce a model for the effective macroscopic behavior through an appropriate averaging, or homogenization, procedure.

Working toward a theoretical approach to homogenization, using a variational principle, and in the context of strain gradient crystal plasticity, we developed a new upper bound for the effective yield surface of a polycrystal.¹¹ This procedure uses one- and two-point correlation functions of the orientation distribution. This bound demonstrates a grain size effect via its dependence on the polycrystal's spatial and orientation distribution statistics. It can be shown that this bound almost always improves upon the Taylor model bound.

It is also possible to approach the homogenization from the point of view of numerical simulations using a so-called *virtual test sample*. A representative volume element of material can be modeled in which individual grains are explicitly resolved, and represented by a single-crystal plasticity model. Such a

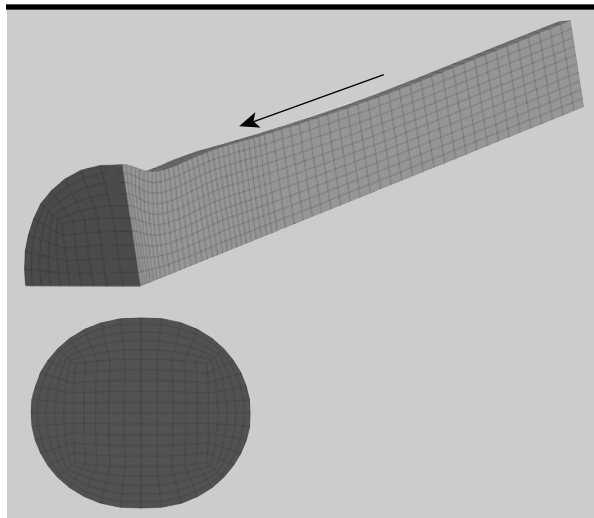


Figure 3. Numerical simulation of a Taylor impact test of an anisotropic tantalum circular cylinder exhibiting ovaling of the impact footprint. For isotropic material properties, the cylinder would expand axisymmetrically at the impacting end. The calculation agrees with experimental results.

numerical test sample can be used to simulate the macroscopic behavior under a wide variety of loading conditions, guiding the development of a macroscopic phenomenological model. In preparation for this approach, detailed orientation imaging microscope (OIM) scans have been made at regular depths through a carefully prepared tantalum specimen, mapping out the orientation of individual grains (Fig. 4). From this detailed database a 3-D numerical model can be generated which accurately represents the real microstructure of the material. This numerical model can then be subjected to a variety of loading conditions, and the effective macroscopic behavior predicted. The yield surface data that has been generated can be used as a validation check for both theoretical and numerical homogenization methodologies.

Future Work

This project has produced an experimental capability that provides multiaxial data, which can be used to develop and validate advanced constitutive models. We have also established a solid framework from which to pursue numerical developments of anisotropic plasticity. From this foundation, further developments will continue, focusing on the development of improved material models for ASCI codes.

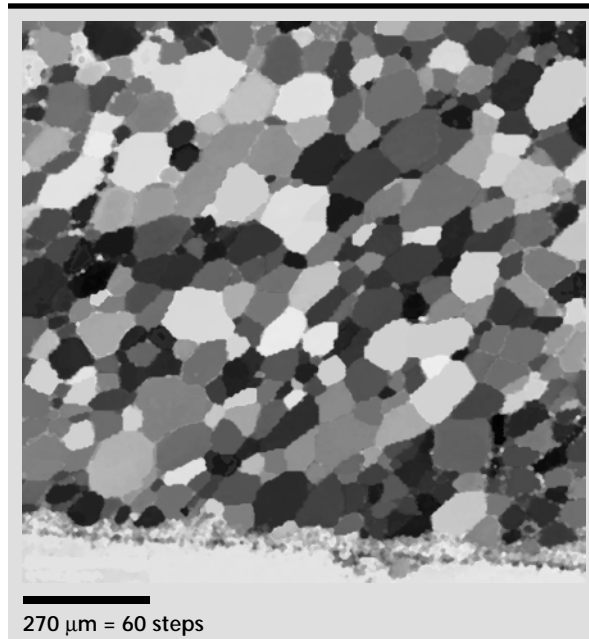



Figure 4. Orientation Imaging Microscope scan of one plane of a well-characterized tantalum polycrystal. The shading indicates particular crystal orientation. Similar scans, at a sampling resolution of $4.5\ \mu\text{m}$, have been made at 47 planes through the sample. From these data a 3-D numerical model can be constructed representing the microstructure.

Acknowledgments

Contributions to this project by J. Casey result from work supported by a grant from the Solid Mechanics Program of the National Science Foundation. Part of the support for A. Brown's work, and material used during the development of the experimental procedure, were also provided through this grant.

References

1. P. M. Naghdi (1990), "A critical review of the state of finite plasticity," *J. Appl. Math. Phys. (ZAMP)*, **41**, pp. 315–394.
2. D. J. Nikkel, A. A. Brown, and J. Casey (1998), "Evolution of anisotropic yield behavior," *Engineering Research, Development and Technology*, Lawrence Livermore National Laboratory, Livermore, California (UCRL53868-97), pp. 5-1–5-5.
3. D. J. Nikkel, A. A. Brown, and J. Casey (1999), "Modeling of anisotropic inelastic behavior," *Engineering Research, Development and Technology*, Lawrence Livermore National Laboratory, Livermore, California (UCRL 53868-98), pp. 6-1–6-6.
4. A. A. Brown, and D. J. Nikkel (1999), "Experimental methodology for yield surface measurement under biaxial states of stress," Lawrence Livermore National Laboratory, Livermore, California, in preparation.
5. J. Casey and P. M. Naghdi (1992), "A prescription for the identification of finite plastic strain," *Int. J. Engng. Sci.*, **30**, pp. 1257–78.
6. J. Casey, and P. M. Naghdi (1992), "On the identification of plastic strain at finite deformation," in *Defects and Anelasticity in the Characterization of Crystalline Solids*, L. M. Brock, ed., ASME AMD Vol. **148**, pp. 11–33.
7. J. D. Lawrence (1972), *A Catalog of Special Plane Curves*, Dover, New York.
8. P. T. Boggs, R. H. Byrd, J. E. Rogers, and R. B. Schnabel (1992), "User's reference guide for ODRPACK version 2.01: software for weighted orthogonal distance regression," U. S. Department of Commerce, (NISTIR 92-4834).
9. A. E. Green and P. M. Naghdi (1965), "A general theory of an elastic-plastic continuum," *Arch. Rat. Mech. Anal.*, **18**, pp. 251–281.
10. P. J. Maudlin, J. F. Bingert, J. W. House, and S. R. Chen (1999), "On the modeling of the Taylor cylinder impact test for orthotropic textured materials: experiments and simulations," *Int. J. Plasticity*, **15**, pp. 139–166.
11. D. D. Sam and V. P. Smyshlyaev (1999), "Grain size effect for the polycrystal's yield surface in strain gradient plasticity," in preparation. 

Development of Laser-Driven Photocathode Injector and Femtosecond-Scale Laser Electron Synchronization for Next Generation Light Sources

Gregory P. Le Sage
Defense Sciences Engineering Division
Electronics Engineering

Thomas E. Cowan
Physics and Space Technology

Todd R. Ditmire
Laser Science and Technology
Laser Programs

J. Rosenzweig
University of California at Los Angeles
Los Angeles, California

A high brightness photoinjector has been developed at Lawrence Livermore National Laboratory (LLNL). This injector, combined with the 100-TW FALCON laser, and the existing LLNL 100-MeV S-Band RF linac will enable development of a high brightness, femtosecond-scale, tunable, hard x-ray probe for time-resolved material measurements. A low emittance electron beam synchronized with femtosecond accuracy to an intense laser will revolutionize x-ray dynamics studies of materials. This project will lead development of ultrafast x-ray dynamics research on problems important in physics, chemistry, biology, and materials science. Precise diagnostics for measurement of relativistic electron beams have also been developed.

Introduction

Short pulse x-rays enable time-resolved characterization of shock dynamics, and examination of materials under extremes of pressure and temperature. Examples include equation-of-state characterization on high-density materials, crystal disorganization and re-growth in shocked and heated materials, and measurement of short time scale phase transition phenomena. Single shot evaluation, requiring high peak flux, is important for complex experiments such as probing of laser-shocked actinides.

Our photoinjector project enables the development of a high brightness, femtosecond-scale, tunable, hard x-ray probe for time-resolved material measurements.

The x-ray pulse intensity will be five orders of magnitude higher than previous demonstration experiments, extending the range of applicability beyond repetitive solid-state applications. Success of this project will place LLNL at the forefront of ultrafast x-ray dynamics and establish a world class electron beam dynamics research program.

The key goals of the photoinjector include development of a photoinjector integrated with the LLNL RF linac, and implementation of techniques to synchronize the photoinjector with an RF linac and the high power FALCON laser system. New techniques have been proposed to meet the synchronization requirements of the system in its final configuration.¹ Characterization of the beam parameters and optimization of the laser-electron interaction will be accomplished using new diagnostic techniques developed at LLNL.^{2,3}

Progress


Despite the highly ambitious goals and schedule, all milestones have so far been met or exceeded. A summary of first year accomplishments follows: (1) high-power RF system commissioned with 18 MW delivered to the photoinjector lab; (2) photoinjector cavity designed, cold tested, brazed, and installed with all specifications met or exceeded; (3) LLNL innovations, including HIP copper, diamond-turned cathode, and improved vacuum; (4) photoinjector test beamline designed and implemented, complete with optics and diagnostics; (5) UV laser system nearing completion; and (6) RF linac synchronized with the FALCON laser system.

Future Work

The first goals for FY-00 include synchronization of the photoinjector and FALCON laser, and continuing detailed characterization of the photoinjector beam. The low energy (4 to 5 MeV) electron beam and the FALCON laser will be used in the early part of FY-00 for Thomson scattering using the photoinjector independent of the RF linac. Photoinjector integration with the linac will follow these initial experiments. Additional experiments relating directly to high gradient accelerator science and technology, as well as advanced diagnostics will be performed with time permitting, and should lead to important publications.

The final product of this proposal enables an operational short pulse x-ray test-bed. Development of the photoinjector and integration with the linac also leads to a world class electron beam research program. The project will evolve into an ongoing program on shocked materials.

References

1. J. P. Rosenzweig and G. P. Le Sage (1998), "Synchronization of sub-picosecond electron and laser pulses," *AIP Conf. Proc.*, No. 472, 795-810.
2. G. P. Le Sage, T. E. Cowan, R. B. Fiorito, and D. W. Rule (1999), "Transverse phase space mapping of relativistic electron beams using optical transition radiation," submitted to *Phys. Rev., Special Topics, Accelerators and Beams*, September.
3. G. P. Le Sage and T. E. Cowan (1999), "A device for transverse mapping of divergence and energy for charged particle beams with arbitrary spatial profiles," Lawrence Livermore National Laboratory, Record of Invention 10561, July 5. 

A Segment-to-Segment Sliding Interface Algorithm for DYNA3D

Jerry I. Lin
Defense Technologies Engineering Division
Mechanical Engineering

This report describes an interface algorithm focusing on the detection of interference among segments, instead of the traditional approach of nodes against segments. It is designed to treat the edge-to-edge contact between structures, but versatile enough to handle general contact/impact problems. This algorithm can be implemented to complement the existing sliding interface algorithms in DYNA3D or as a stand-alone sliding interface option.

Introduction

There are several sliding interface options available in the explicit finite-element code DYNA3D.¹ They are designed to address different contact/impact scenarios and are based on the detection of interference between a node and a segment. A segment in sliding interface terms usually means a shell/plate element or a facet of a volume element. Even for the contact options with both sides of the sliding interface defined by segment lists, DYNA3D internally sorts out the nodes associated with one segment list and checks against the other segment list. The node-to-segment approach hinges on the detection of the interference between a node trajectory within a certain period of time, usually a time step, and a given segment. The trajectory of a node within a integration time step can be properly represented by its velocity vector. This method has been the backbone of the DYNA3D sliding interfaces and has performed very well over time, but it fails to detect the interference for certain cases.

One of these cases is the so-called edge-to-edge contact that is depicted in **Fig. 1** in an extreme way. The arrow in the figure indicates the motion of one shell element; the other is stationary. The two shell elements clearly interpenetrated each other, but will not be detected because the trajectories of the nodes associated with one element never interfere with the other element.

Experienced finite-element analysts have used modeling techniques to overcome this shortcoming in simulating edge-to-edge contact. One of the methods used is to add rigid shell elements perpendicular to the shell structure edges to aid the contact

detection. However, modeling techniques of this nature require knowing the penetrating entities *a priori* and often lead to falsely identified interference.

For analyses involving unpredictable structural damage, such as the damage assessment of bomb-blasted buildings or the integrity assessment of underground structures, these modeling techniques appear impractical since the disintegration of structures is unknown to the analyst until the event takes place.

The failure of the nodes-to-segments sliding interface algorithms can be attributed to the loss of sight of segment connectivity on the side that is represented by a simple list of nodes. To remedy this problem, a segment-to-segment algorithm, which considers the nodal connectivity of both interfering entities, is a logical alternative. Because of the complicated mathematical representation for a polygon in 3-D space, a definitive identification of segment interference can be costly. To make methods of this class feasible in a production finite-element code like DYNA3D, robust and effective procedures in checking the interference must be

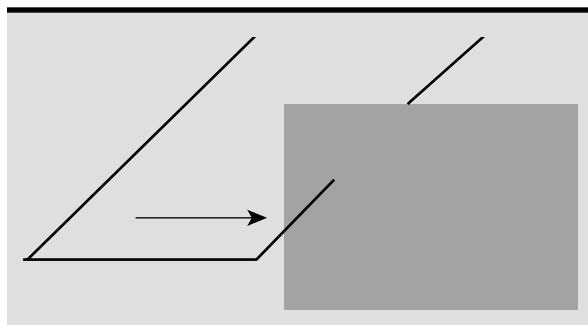


Figure 1. Segment edge-to-edge contact.

made available. A proposed detection algorithm that does not require solving nonlinear equations is outlined in the following section. With the help of the existing sorting² and preliminary screening³ techniques, this segment-to-segment detection method represents a robust and complementary alternative to the node-to-segment approaches. Once the interference between a pair of segments is identified, it can be resolved by applying the separation forces to either the segment or the individual nodes.

Progress

The Segment-to-Segment Detection Procedure

Because of the significant cost associated with the contact detection calculation, all sliding interface algorithms, including node-to-segment and segment-to-segment, prefer to deal with as few interfering candidates as possible. Two preliminary screening techniques, namely the bucket sort² and the pinball search,³ are frequently used in reducing the candidate list. Both of these methods are implemented in the existing DYNA3D sliding interface options and over the years have proven to be highly effective. They will again be used in the proposed detection scheme to sort out the possible interfering pairs. The segment-to-segment detection procedure is described in its execution sequence as follows:

1. The Bucket Sort. An imaginary rectangular parallelepiped large enough to cover the entire contact domain is imposed on the sliding interface. This box is then broken into contiguous cubes, originally called *cells*³ but more commonly referred to as *buckets*. The size of these buckets is determined by the code. All relevant segments are sorted into these buckets according to the locations of their centroids. When detecting the interference for a specific segment, only segments located in the same or adjacent buckets will be checked against it in the subsequent steps.
2. The Pinball Search. A sphere, hence the name *pinball*, is assigned to each segment on the sliding interface. The pinball is centered at the centroid of the segment and is large enough to enclose its associated segment. A preliminary screening of identifying overlapping pinballs is conducted next. Only pairs of segments with overlapping pinballs will be further examined in the subsequent steps.

3. The Segment Interference Check. Segments *A* and *B* in **Fig. 2**, defined by nodes 1-2-3-4 and 5-6-7-8 respectively, represent a generic pair being checked for possible interference. \mathbf{n}_A and \mathbf{n}_B are their unit normal vectors, and \mathbf{x}_A and \mathbf{x}_B are the locations of their centroids. (Bold face characters denote vector quantities in the rest of our discussion.) Side 5-6 of segment *B* is first checked against segment *A*. Let us denote the plane passing through \mathbf{x}_A and having \mathbf{n}_A as its unit normal vector plane *A*. The following parametric equation of *p* represents the point at which the ray 5-6 intersects plane *A*.

$$[\mathbf{x}_5 + p(\mathbf{x}_6 - \mathbf{x}_5) - \mathbf{x}_A] \cdot \mathbf{n}_A = 0 \quad (1)$$

Solving for *p* yields

$$p = \frac{(\mathbf{x}_A - \mathbf{x}_5) \cdot \mathbf{n}_A}{(\mathbf{x}_6 - \mathbf{x}_5) \cdot \mathbf{n}_A} \quad (2)$$

If $0 \leq p \leq 1$, which means nodes 5 and 6 are on opposite sides of plane *A*, the intersection point *P* can be expressed as

$$\mathbf{x}_P = \mathbf{x}_5 + p(\mathbf{x}_6 - \mathbf{x}_5) \quad (3)$$

Otherwise side 5-6 of segment *B* does not intersect segment *A*, and no further action is needed for side 5-6. For the case $0 \leq p \leq 1$, we must determine whether \mathbf{x}_P falls within segment *A*. If

$$(\mathbf{x}_P - \mathbf{x}_3) \cdot [(\mathbf{x}_2 - \mathbf{x}_3) \times \mathbf{n}_A] \geq 0, \quad (4)$$

\mathbf{x}_P is on the same side of line 2-3 as the segment centroid \mathbf{x}_A . If the inequality (**Eq. 4**)

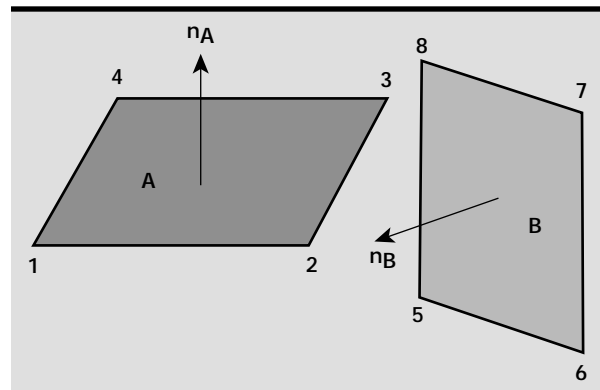


Figure 2. A pair of segments under interference check.

holds true, the same check must be applied to the remaining three sides of segment A. If any of the four sides of segment A do not satisfy Eq. 4, side 5-6 of segment B does not intersect segment A, and we should proceed to check the remaining sides of segment B. Only if \mathbf{x}_p is on the same side as segment centroid \mathbf{x}_A for all four sides of segment A, can the interference be declared. Once interference with segment A is identified for any one side of segment B, the unchecked side(s) of segment B can be skipped.

4. The Centroid Position Check. One scenario of interference, as shown in Fig. 3, can still go undetected by the procedures outlined in step 3. In this case, the two segments under consideration are perfectly parallel and the subsequent motion does not change this situation. Even when the two segments pass through each other, the sides of a segment do not intersect its counterpart. However this situation can be quickly identified by

$$(\mathbf{x}_B - \mathbf{x}_A) \cdot \mathbf{n}_A \leq 0, \quad (5)$$

which indicates \mathbf{x}_B , the centroid of segment B, has dipped under segment A.

Interference Resolution

After the interference between two segments is detected, a set of forces must be applied to resolve the interference. These forces can be decided either by examining the relative position and motion between individual nodes and the opposing segment or by examining the relative position and motion of two segment centroids.

The former is essentially going back to a localized node-to-segment approach. It allows forces of different magnitudes and directions to be imposed on individual nodes according to their respective position and motion. Its shortcomings are higher cost,

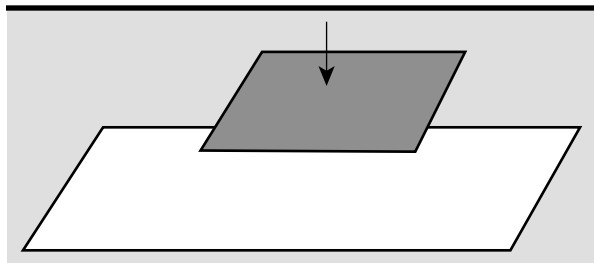


Figure 3. Two parallel segments: the top one passes through the bottom one.

increased possibility of erroneous nodal motion, and the need for double-path processing to maintain possible symmetry.

The latter approach treats the segments as collective entities and calculates the separation forces as functions of relative position and motion. Nodes are not examined individually, and forces are applied uniformly to the nodes of a segment. This approach may not be adequate for cases requiring accurate node repositioning. However it will be more cost-effective, stable, and inherently maintains possible symmetry.


Future Work

It is very difficult, if not impossible, to design a sliding interface algorithm that covers all contact conditions. The proposed detection scheme complements the existing sliding interface options in DYNA3D and makes analyzing problems with arbitrary structure disintegration feasible. This new algorithm does not incur great expense since no iterations or solving systems of equations is required. There should be little doubt that extreme scenarios can be drawn up to foil the algorithm, but it should be able to simulate most of the contact situations in our application.

Further studies and experiments are needed for the separation force calculation. This is especially true if a node-to-segment approach is to be adopted and individual node trajectories do not intersect the opposing segment. Past research in this area³ should provide a good guideline for future work.

The implementation of this algorithm into DYNA3D is to be carried out in the next fiscal year, and the new feature should be ready by the latter half of 2000.

References

1. R. G. Whirley and B. E. Engelmann (1993), "DYNA3D: a nonlinear, explicit, three-dimensional finite element code for solid and structural mechanics—user manual," Lawrence Livermore National Laboratory, Livermore, California (UCRL-MA-107254).
2. T. Belytschko and J. Lin (1987), "A three-dimensional impact-penetration algorithm with erosion," *Computers and Structures*, Vol. 25, pp. 95–104.
3. T. Belytschko and M. O. Neal (1991), "Contact-impact by the pinball algorithm with penalty, projection and Lagrangian methods," *Int. J. for Numer. Methods in Eng.*, Vol. 31, pp. 547–572. 



Usability Enhancements for TIGER

David J. Steich, Gerald J. Burke, and Jeffery S. Kallman
Defense Sciences Engineering Division
Electronics Engineering

S Terry Brugger
Computer Applications Sciences and Engineering
Computation Organization

Daniel A. White
Center for Applied Scientific Computing
Computation Organization

This report discusses our progress with the Time-Domain Generalized Excitation and Response (TIGER) code. Our activities this year focused on building usability enhancements for TIGER and modeling a septum structure.

Introduction

Last year we extended our prototype version of the TIGER code to model realistic accelerator kicker structures being built for the Experimental Test Accelerator (ETA-II) at Lawrence Livermore National Laboratory (LLNL). We also added high-order wake field capabilities for the kicker structures and explored time-domain thin-wire algorithms.

This year we began by modeling a septum to better understand the wake fields caused by having the high-current electron beam pass by this structure. The modeling of the septum, shown in **Fig. 1**, presented many new challenges. Although the septum has many similarities with kicker structures, there are also significant physics differences.

This is the last year we will build on the prototype version of TIGER. The prototype version allowed us to test many of our object-oriented abstractions in TIGER while solving some near-term problems of interest to LLNL's accelerator programs. It is our intent to move the prototype version's capabilities into a more production level code. The work involved in building a production level, time-domain, full-wave physics electromagnetics code for structured/unstructured, serial/parallel situations involves two main efforts.

The first effort is the management of the complexity and the bookkeeping involved in hybrid

meshes, massively parallel environments, and multiple algorithms with disparate overheads. The second effort is the research into the generalization of algorithms so they work in hybrid situations. Both of these efforts are non-trivial. It is our hope that the object-oriented framework that we have been building will insulate the programmer from the complexities of hybridization involved in the first task. At the same time, we hope the platform provides a programming environment that facilitates quick testing and implementation of the new physics that must be developed for the hybridizations/generalizations of the second task.

The object-oriented framework is several months behind schedule, but is proceeding rapidly. A first beta version of a series of three libraries, which represent the core of TIGER's object-oriented bookkeeping abstractions, is intended to be finished in early 2000. Earlier versions of TIGER have, to varying degrees, tested major portions of the abstractions. The latest version ties everything together.

Progress

This year's efforts were directed at TIGER usability enhancements and the modeling of a septum structure. Specifically, we built a simple Graphical User Interface (GUI) and mesh generator, developed and implemented a general field sensor capability

for unstructured finite volume meshes, developed and implemented a current sourcing procedure for unstructured grids, and ran several septum applications requiring these features.

These usability enhancements were required to help transition TIGER's research technologies into a more production level environment. The development of sourcing and sensing capabilities for unstructured meshes, described in this report, were unresolved research topics that needed to be addressed to increase TIGER's overall applicability.

Meshing and GUI

The GUI we built is at present very limited, but does allow us to query the mesh about detailed topological questions. The mesh generator is simplistic and was built solely to circumvent difficulties encountered using the I-DEAS mesh generation package. Our mesh generator builds an all-hexahedral mapped mesh.

Although the mesh generator is very crude and not user friendly, it allows us to build septum geometries in a few hours to a few days depending on the complexity of the geometry. We were still not quite finished building the septum shown in **Fig. 1** after spending roughly two months of effort using the I-DEAS software package. The knife edge seam where the waveguide section splits into two rectangular pipes, as well as the curved depression areas in the waveguide, caused considerable difficulties in I-DEAS. **Figure 2** shows part of the outer surface mesh of the septum built with our mesh generator and displayed using the GUI.

3-D Vector Field Interpolation

The finite-difference time-domain (FDTD) algorithm is a simple but elegant algorithm for modeling Maxwell's equations. The algorithm has matured since its inception at LLNL in 1966. FDTD is a very powerful computational method and is still by far the most popular method used today. FDTD does have a severe limitation in that the modeled geometry must be built out of rectangularly shaped brick elements.

Finite-volume time-domain (FVTD) techniques such as the discrete surface integral (DSI)¹ method remove this limitation by allowing the mesh to conform to the geometry. However, FVTD techniques are much newer, more complex, more research-oriented, and are less mature, with only dozens of papers published to date as compared to thousands for FDTD.

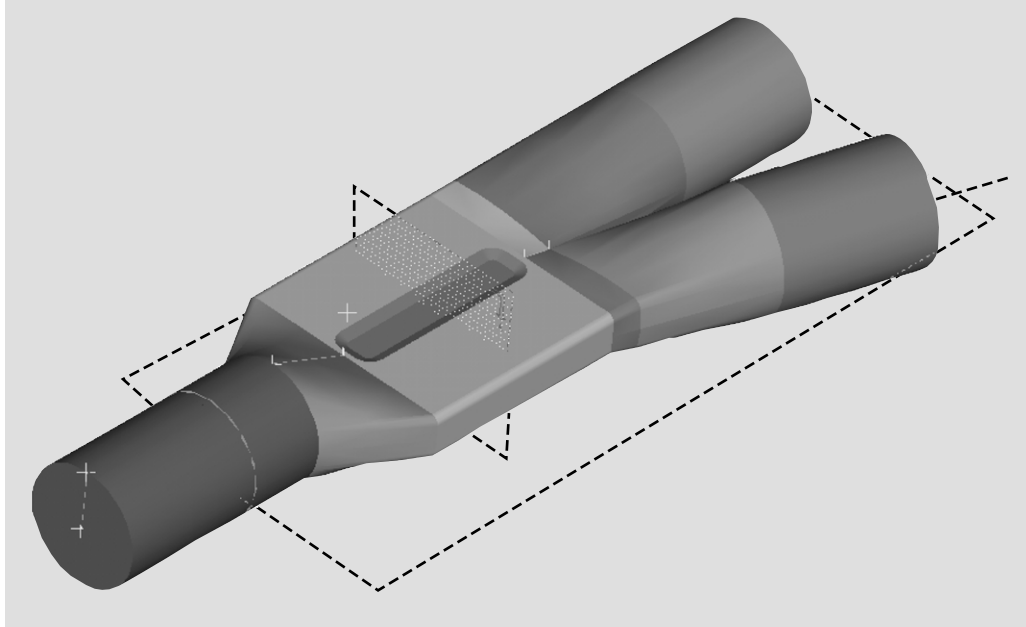
One open research area for FVTD techniques is the implementation of sources and sensors when the mesh is non-orthogonal. In a FDTD method, the mesh is always orthogonal and so each of the vector field components is de-coupled from the others.

This allows one to interpolate/extrapolate various field components independently.

In a finite-element method (FEM), basis functions are used which uniquely determine the fields at any point within the element. In a FVTD method however, the vector field components are coupled when the mesh is non-orthogonal, and in general, no basis function has been assumed to determine fields at an arbitrary point.

When modeling accelerator components it is often desirable to compute wake potentials by integrating

Figure 1. Septum structure modeled to determine wake potentials.



fields over an arbitrary cylindrical surface that does not conform to the mesh.

Usually this surface is determined by the physics, and cannot conform to the mesh. Since the edges in the unstructured mesh do not conform to the cylindrical surface, and in general are not aligned with the coordinate axes, the fields must be interpolated to the required points. Linear interpolation was used on the vector fields in three dimensions. The vector interpolation function for point (x, y, z) is

$$\mathbf{F}(x, y, z) = F_x(x, y, z)\mathbf{x} + F_y(x, y, z)\mathbf{y} + F_z(x, y, z)\mathbf{z}$$

with

$$F_x(x, y, z) = C_1 + C_2x' + C_3y' + C_4z' + C_5x'y' + C_6x'z' + C_7y'z' + C_8x'y'z' \quad (1)$$

$$F_y(x, y, z) = C_9 + C_{10}x' + C_{11}y' + C_{12}z' + C_{13}x'y' + C_{14}x'z' + C_{15}y'z' + C_{16}x'y'z' \quad (2)$$

$$F_z(x, y, z) = C_{17} + C_{18}x' + C_{19}y' + C_{20}z' + C_{21}x'y' + C_{22}x'z' + C_{23}y'z' + C_{24}x'y'z' \quad (3)$$

and $x' = x - x_0$, $y' = y - y_0$, $z' = z - z_0$ for a reference point (x_0, y_0, z_0) . To determine the constants C_j a set of 24 edges is chosen in the vicinity of (x_0, y_0, z_0) and the dot products of $\mathbf{F}(x, y, z)$ with the edge directions are matched to the edge fields from the DSI solution. If d_{ix} , d_{iy} and d_{iz} are the direction cosines of edge i , and E_i is the edge field at (x_i, y_i, z_i) , the

set of 24 equations is

$$E_i = \mathbf{F}(x_i, y_i, z_i) \cdot \mathbf{d}_i = \quad (4)$$

$$(C_1 + C_2x'_i + C_3y'_i + C_4z'_i + C_5x'_iy'_i + C_6x'_iz'_i + C_7y'_iz'_i + C_8x'_iy'_iz'_i)d_{ix} + (C_9 + C_{10}x'_i + C_{11}y'_i + C_{12}z'_i + C_{13}x'_iy'_i + C_{14}x'_iz'_i + C_{15}y'_iz'_i + C_{16}x'_iy'_iz'_i)d_{iy} + (C_{17} + C_{18}x'_i + C_{19}y'_i + C_{20}z'_i + C_{21}x'_iy'_i + C_{22}x'_iz'_i + C_{23}y'_iz'_i + C_{24}x'_iy'_iz'_i)d_{iz}$$

for $i = 1, \dots, 24$. This set of equations is solved for the coefficients C_j to determine the interpolation function. If A_{ij} are the elements of the inverse of the matrix of coefficients of the C_j in **Eq. 4**, then

$$C_i = \sum_{j=1}^{24} A_{ij}^{-1} E_j$$

for $i = 1, 24$.

In evaluating the wake potentials we interpolated the fields to points with spacing on the order of the mesh cell dimensions and approximated the integrals as sums of pulses. In this case, each interpolation function and corresponding set of edges is used for only one evaluation point. By choosing the reference point at the evaluation point x' , y' and z' become zero, and the interpolation function simplifies to

$$\mathbf{F}(x_0, y_0, z_0) = \sum_{j=1}^{24} (A_{1,j}^{-1}\mathbf{x} + A_{9,j}^{-1}\mathbf{y} + A_{17,j}^{-1}\mathbf{z})E_j$$

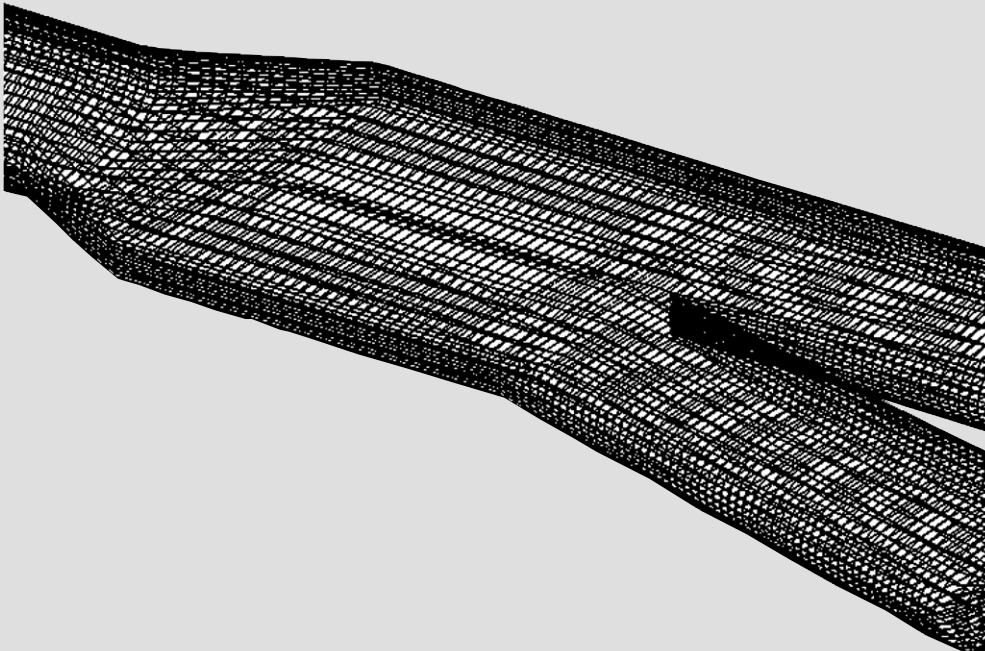


Figure 2. Mesh developed for the septum model.

Thus only 3×24 coefficients must be stored for each point for use with the E_i from each time step.

The choice of edges for fitting the interpolation function was found to require care, since a bad choice could result in a singular or ill-conditioned matrix. For each interpolation point all edges within a radius of several times the maximum cell dimension were selected. These were separated into edges predominately in the x , y , and z directions and sorted by distance from the interpolation point. Then for each vector direction the nearest edges were chosen, to obtain eight edges surrounding the point in the x , y , and z directions. When an edge in a given octant of space about the point could not be found, the next nearest edge in another octant was used, but this sometimes resulted in an ill-conditioned matrix. This process worked smoothly once the division between x -, y -, and z -directed edges had been optimized for the meshes being used, although fancier logic could make the process more robust.

Non-Orthogonal Current Source Modeling

The 3-D vector interpolation works for sensing the fields at arbitrary points in a non-orthogonal mesh. The scheme works perfectly, neglecting finite precision errors, for fields that have a linear variation. One must be more careful when dealing with sources. A rigid electron beam with known trajectory can be modeled as a current source in a FVTD scheme. However, the current source must be introduced so that it is charge conserving and causal.

Using FDTD or FVTD on orthogonal grids for current sources moving along straight grid edges is trivial. The implementation of current sources on orthogonal grids is still straightforward even if the current sources change direction, as long as the above mentioned constraints are enforced and the current sources remain tied to grid edges. The left side of **Fig. 3** depicts typical DC fields left behind if one models current jumping from one edge to

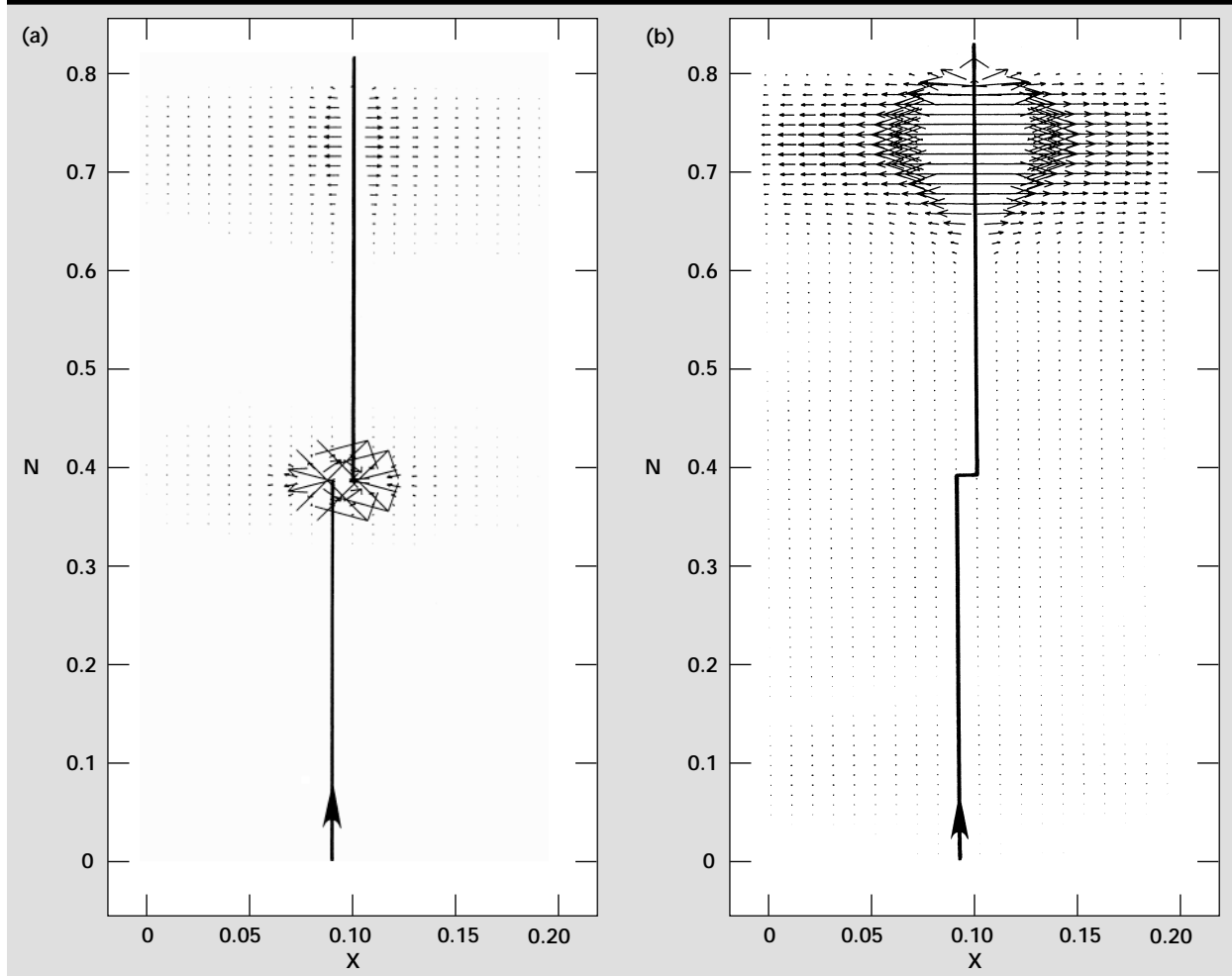


Figure 3. Fields produced by a current source moving at the velocity of light, showing the DC fields that result when continuity and causality are not correctly enforced (a) and the correct result (b).

another instantaneously without changing direction on a continuous path. The right side of **Fig. 3** depicts the correct way of modeling sources in a FDTD method. The current path is continuous around the stair-steps and causality is enforced by appropriate edge-to-edge delays in the incident current. The large DC field left behind when continuity or causality are not enforced correctly can often produce late time instabilities in a FDTD or FVTD method.

Now consider how to model a current source that was traveling just a few degrees tilted with respect to one of the grid axes so that it cannot conform to the grid edges. One could model the current source as a superposition of current sources along surrounding grid edges, weighted based on the local source position. This approach causes the beam cross-section to increase significantly.

One could model the current source as in the right side of **Fig. 3** where the current source trajectory is a series of edges that are the closest fit to the true path. This approach works well unless high-frequency phase information is critical. One could surrender and use a full-blown Particle In Cell (PIC) approach at the expense of dramatically increasing memory and CPU overheads. Or lastly, one could modify the mesh to conform to the source trajectory. In general, this will force the mesh surrounding the sourced edge to be non-orthogonal where the field components are coupled. This latter approach is the one we have taken. The modification of the DSI algorithm to introduce a current source through the cells is outlined below.

Consider updating a current edge as shown in **Fig. 4**. Using the DSI method for the unstructured mesh, the change in the electric flux density at the

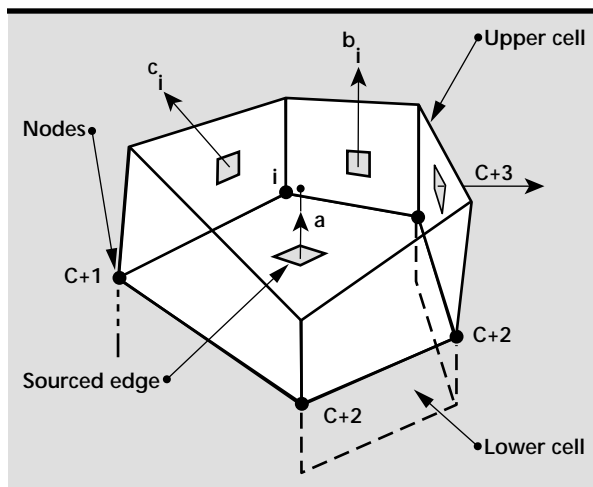


Figure 4. Mesh faces used in updating a field component in the DSI algorithm.

corner of faces a, b_i, c_i is

$$\frac{\partial D_i}{\partial t} = \frac{b_i \times c_i \sum_a + c_i \times a \sum_{b_i} + a \times b_i \sum_{c_i}}{a \cdot b_i \times c_i}$$

where for each face a, b_i, c_i we define

$$\oint_{\text{Face}} \vec{H}^k \cdot d\vec{l} \equiv \sum_{\text{Face}}$$

and a, b_i, c_i are vector area normals for the faces depicted in **Fig. 4**. Assuming we are one or more cells away from any exterior boundaries, the full electric flux density is the summation of the partial electric flux densities traversing the corners of face a for the upper and lower cells. The total electric flux density can then be written as

$$\frac{\partial D_{\text{total}}}{\partial t} = \sum_{\text{cell}}^{\text{upper+lower}} \sum_i w_i \frac{\partial D_i}{\partial t}$$

where w_i is a weighting term.

The electric field dotted along a given edge can then be easily written using the above equation. However, correction terms must also be added to every electric field update that uses face a in its update stencil. In simple terms, every \sum_a should be replaced with $\sum_a^{+I_a}$. The electric field for a given edge including all potential current sources can be written

$$E^k \cdot d\vec{l} = E^{k-1} \cdot d\vec{l} + \frac{\Delta t}{\epsilon} \left(\sum_{\text{cell}}^{\text{upper+lower}} \frac{\sum_i w_i \left(b_i \times c_i \left(\sum_a + I_a \right) + c_i \times a \left(\sum_{b_i} + I_{b_i} \right) + a \times b_i \left(\sum_{c_i} + I_{c_i} \right) \right)}{\sum_i a \cdot b_i \times c_i} \right) \cdot d\vec{l} \quad (5)$$

where I_{b_i} and I_{c_i} are current sources through faces b_i and c_i respectively. In general, there will be eight correction terms for each sourced edge on a hexahedral grid.

The first term on the right hand side of **Eq. 5** can be thought of as the orthogonal portion while the remaining terms on the right hand side can be thought of as correction terms due to the non-orthogonal mesh. These non-orthogonal corrections are required to preserve current continuity. Non-physical DC fields and instabilities can arise when these correction terms are missing. In a fully Cartesian mesh **Eq. 5** reduces to the usual update equation including current.

Equation 5 leaves no DC fields behind. It is noteworthy to mention that even if the correction terms are very small, say $\sim 10^{-4}$ of the orthogonal contribution, their contribution can dramatically affect the

overall solution. Even changing the contributions in the sixth or seventh decimal place can have a significant impact on the overall solution accumulated over time. Research into why these terms are overly sensitive to numerical precision is ongoing.

Septum Modeling

Figure 5 shows the outer surface of the septum built for the ETA-II accelerator. **Figures 1 and 2** show the actual septum we modeled. The purpose was to get a better understanding of the wake fields caused by such a structure. A high-current electron beam is “kicked” upstream. The beam comes into the septum waveguide structure displaced ± 3 cm off the center axis of the septum. The beam has a tilt of roughly 0.86° entering the waveguide section.

In computing and processing the wake fields we started with the septum model and later looked at simpler situations such as a “T” junction in the pipe. The septum presented a number of new problems that made the wakefield processing more difficult than in previous cases such as the kicker. With the kicker, or any straight pipe that does not have protrusions into the pipe volume, the beam source can be replaced by equivalent source fields along the pipe walls. The equivalent source fields exist only over cavities or apertures in the pipe walls, so errors due to launching the wave from the problem-space boundary are avoided, and the source does not accumulate dispersion error over the length of the pipe. Early modeling of wake fields with TSAR did not use equivalent sources, but the solution for the field of the beam in a clean pipe was subtracted from the solution for the pipe with cavity or kicker.

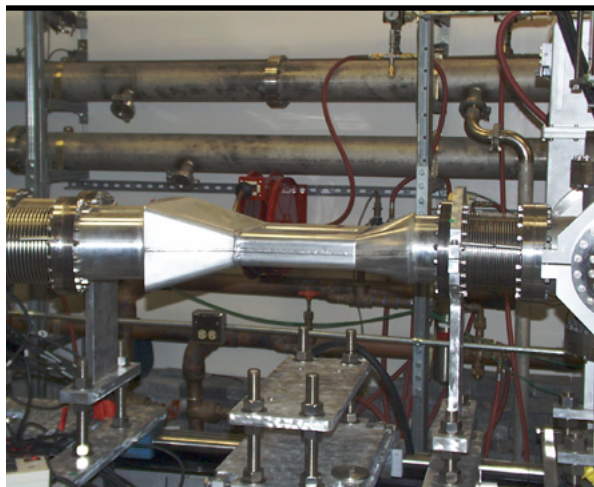


Figure 5. Outer surface of the ETA-II septum.

While errors were introduced in launching the source field and through dispersion, the subtraction was very effective in canceling these errors. Neither an equivalent source nor subtraction could be used with the septum, since there is no straight pipe path through the structure. Hence we had to work with the raw fields from the solution for the septum model.

Since the septum was modeled with the DSI algorithm, a continuous string of edges could be chosen to follow the beam path. The source was introduced as a Gaussian pulse of current traveling at the velocity of light, $v = c$, along the source edges. Fields were set up over the boundary where the pulse entered the problem space as if the pulse had come from infinity. The fields set up on the entry surface and also the absorbing boundary introduce errors into the solution, but these errors are confined to near the boundaries and were excluded in computing wake fields.

As each model was run, the electric and magnetic fields were saved for all edges and all time steps. Then in post-processing the fields were selected along the beam path or on a cylinder about the path. The fields on the cylinder were evaluated for 32 azimuthal angles about the beam using linear vector interpolation. Wake potentials were computed from the fields using the standard definitions

$$W_z(r, s) = \frac{1}{Q} \int_{z=-\infty}^{\infty} E_z(r, z, t) \Big|_{t=(z+s)/v} dz$$

$$W_t(r, s) = \frac{1}{Q} \int_{z=-\infty}^{\infty} (E_r(r, z, t) - vH_\phi(r, z, t)) \Big|_{t=(z+s)/v} dz$$

The wake potentials were then Fourier-transformed to obtain coupling impedances. When the wake potentials had significant ringing at the end of the solution record they were extrapolated using

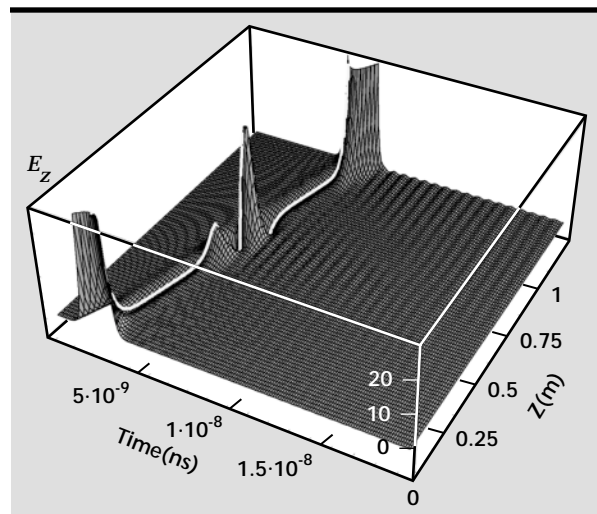


Figure 6. Longitudinal field along the beam path through a “T” junction. The velocity of light line representing the source pulse is shown as a white line.

decaying exponentials obtained with the Generalized Pencil of Functions method.²

Before computing the wake potentials we examined the longitudinal $E_z(r, z, t)$ fields and transverse field $E_r(r, z, t) - vH_\phi(r, z, t)$ on 3-D plots versus z and t . A typical plot is shown in **Fig. 6** for a simple “T” pipe junction. The wake potential $W(s)$ is the integral of the field along a line $t = (z+s)/v$, where $s = 0$ corresponds to the beam line shown in white in **Fig. 6**. Plots such as **Fig. 6** show the sources of the wake fields and can also reveal errors that would go unnoticed in the final wake potential. A number of problems were corrected in this way before getting the final wake potential results. The large fields near the ends of the beam line are errors due to launching and absorbing the pulse and were excluded from the fields used in the wake potential evaluation. The crease following the beam line is due to inherent solution and processing errors. It can be removed when a solution for a clean pipe can be subtracted. Otherwise it must be considered as a source of error in the wake potential, since it has low amplitude but long duration.

A plot of the field E_z along a line in the z, t space of **Fig. 6** is shown in **Fig. 7** for the $m = 0$ mode of the septum. The dotted line is the field $E_z(0, z, z/v)$ normalized to fit on the plot, and the solid line is the integral of this field. Thus the final value of the solid line represents the wake potential $W_z(0)$. This plot shows the scattered field due to the split in the septum where the waveguide contracts in height and expands in width before splitting. The field, and hence the kick imparted to the particle, has both

positive and negative components that partially cancel in the integral.

The wake field in the input pipe has a small value, but makes a significant contribution to the integral due to its long duration. This field may be partly due to inherent errors in the solution and processing and partly due to the real effect of the beam traveling on a tilted path with respect to the pipe axis. One method considered to reduce the processing errors was to start the beam parallel to the axis and then bend it to the correct path. The field at a point where the beam is parallel to the axis can then be used to remove some solution and processing errors over the rest of the beam path. While not as effective as subtracting a full clean-pipe solution, this method reduces the errors somewhat. The standard evaluation of wake potential, with integration to infinity, would require that the beam start and end parallel to the pipe axis, but that raises questions of how to treat the fields produced where the beam changes directions without a self-consistent PIC solution. In our modeling we considered only a tilted beam truncated about 0.6 m before and after the split in the septum. While the result obtained will depend on the truncation distances, our modeling showed the wake potential of the septum to be much smaller than that of the kicker, and on the order of that for the “T” junction discussed below. Experimental results appear to verify this assessment.

To get a better understanding of the wake fields in a structure having some of the properties of the septum we modeled several structures such as the

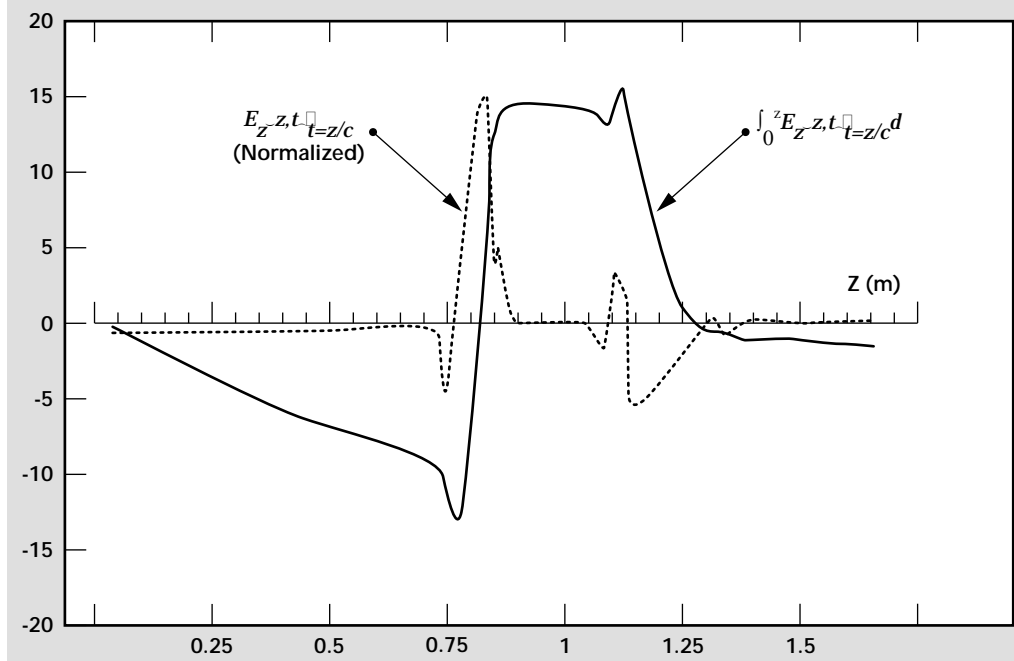
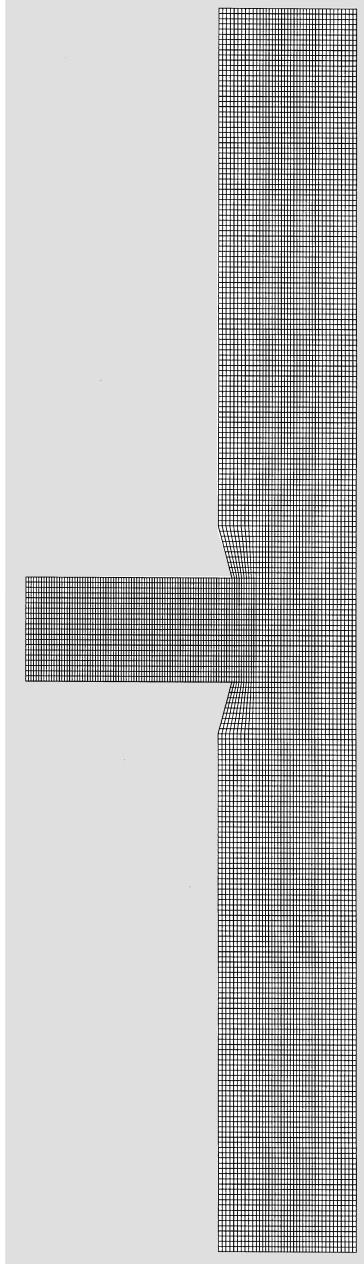


Figure 7. Longitudinal wake field and the integral for the wake potential $W(s)$ at $s = 0$ for the septum. The scale applies for the wake potential integral, while the wake field has been normalized.

“T” junction shown in **Fig. 8**. The notch in the beam pipe in **Fig. 8** is a result of the side pipe having straight edges on its end. We also modeled smooth pipe junctions. The “T” junction is similar to the DHART septum, but with the branch normal to the beam pipe rather than tilted. Since this model had a straight pipe, the clean-pipe solution could be subtracted to reduce errors. The longitudinal wake impedance for the “T” is shown in **Fig. 9** for $m=0, 1$ and 2 , and transverse impedances for $m=1$ and 2 are shown in **Fig. 10**. It should be noted that m indicates the azimuthal variation of the wake impedance, but an $m=0$ source is always assumed here.

Figure 8. Section through the mesh for a “T” junction with the side pipe ending in straight edges.



The coupling impedances for the “T” pipe were compared with approximations based on scattering by a small hole in a pipe published by Gluckstern.³ Gluckstern derived the $m=0$ longitudinal impedance due to a $m=0$ source and the $m=1$ transverse impedance due to a $m=1$ source. Since we considered only $m=0$ sources here, we followed Gluckstern’s small-hole analysis to get the longitudinal impedance approximations for $m=0, 1$ and 2 :

$$Z_{z0}(k) = \frac{ja^3 k \eta_0}{6b^2 \pi^2}$$

$$Z_{z1}(k) = \frac{ja^3 k r \eta_0}{3b^3 \pi^2}$$

$$Z_{z2}(k) = \frac{ja^3 k r^2 \eta_0}{3b^4 \pi^2}$$

and transverse impedance approximations

$$Z_{x1}(k) = \frac{ja^3 \eta_0 \cos(\theta)}{3b^3 \pi^2}$$

$$Z_{x2}(k) = \frac{2ja^3 r \eta_0 \cos(2\theta)}{3b^4 \pi^2}$$

for pipe radius a , hole radius b and test particle at r . Gluckstern’s analysis assumes a hole small compared to the pipe diameter and wavelength and a thin pipe wall. For an infinitely thick pipe wall, corresponding to our case, he suggests a correction factor of 56% of the above values. For **Figs. 9** and **10**, the hole in the beam pipe had width 8 cm and length 10 cm, so we used $b = \sqrt{lw/\pi} = 0.05$ m, while $a = 0.0667$ m. With the 56% correction these approximations yield 2.16 Ω , 0.64 Ω and 0.097 Ω for longitudinal impedance at 1 GHz for $m=0, 1$ and 2 , respectively, and 3.09 Ω and 0.93 Ω for transverse impedance for $m=1$ and 2 . Considering the large

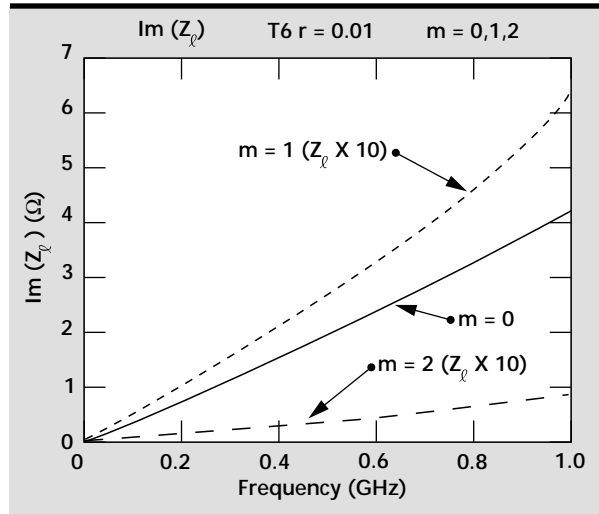


Figure 9. Longitudinal wake impedance for a “T” junction in a beam pipe, showing the $m=0, 1$, and 2 response to a $m=0$ beam excitation.

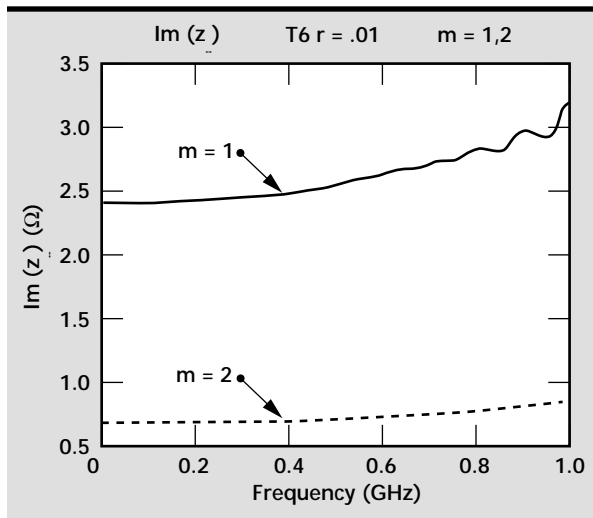


Figure 10. Transverse wake impedance for a "T" junction in a beam pipe, showing the $m = 1$ and 2 response to a $m = 0$ beam excitation.

diameter side pipe in our model violates some of Gluckstern's assumptions, these values are in reasonable agreement with the results in **Figs. 9** and **10**, with the largest error occurring for $m = 0$ longitudinal. It appears that the wake fields of this "T" structure are on the order of those of the septum.

We also modeled a "T" junction where the hole in the beam pipe was a slot 2 cm wide by 20 cm long. The longitudinal $m = 0$ impedance at 1 GHz was 0.8 Ω , in agreement with Gluckstern's assertion that slots have lower impedance than circular holes of similar area. A "T" model with a smooth beam pipe junction showed a somewhat smaller wake potential than the model of **Fig. 8**. We also ran 2-D cases involving a pipe with a step in, step out, cavity or collimator. We compared the results to analytic solutions and performed symmetry checks (a dual "T") and convergence checks to gain confidence.

In summary, we can obtain accurate wake fields for situations where we can subtract a solution for a clean pipe, or any reference solution for zero wake


potential. If wake fields are very small (10^{-4} to 10^{-5} down from the source fields) then numerical errors can become a dominant error source. When the beam is tilted in the pipe, modified definitions for wake potentials must be developed before more meaningful results can be obtained.

Future Work

We plan on incorporating the new physics and usability enhancements developed here into a more production level code. All of these enhancements, along with the developments of previous efforts are vital as we transition TIGER's technologies out of a research environment and into a user environment.

As always, there are many things on our wish list to incorporate into this new code. However, at the top of the list, is the need to more accurately model highly distorted non-orthogonal cells near boundaries. Currently, when we have distorted cells next to metal surfaces we can have a small charge buildup. This charge buildup can lead to errors and instabilities. We have seen this charge buildup near the knife edge seam of the septum, especially when the cells are highly distorted.

References

1. N. K. Madsen and R. W. Ziolkowski (1988), "Numerical solution of Maxwell's equations in the time domain using irregular nonorthogonal grids," *Wave Motion*, Vol. **10**, pp. 583-596.
2. T. K. Sarkar and O. Pereria (1995), "Using the matrix pencil method to estimate the parameters of a sum of complex exponentials," *IEEE Antennas and Propagation Magazine*, Vol. **37**, No. 1, February.
3. R. L. Gluckstern (1992), "Coupling impedance of a single hole in a thick-wall beam pipe," *Phys. Rev. A*, Vol. **46**, No. 2, July 15. 



Enhanced Fluid Dynamics Capability and Multidisciplinary Coupling in ALE3D

Rose C. McCallen, Timothy A. Dunn, and John S. Dolaghan
New Technologies Engineering Division
Mechanical Engineering

Richard G. Couch
New Technologies Engineering Division
Defense and Nuclear Technology
Mechanical Engineering

An incompressible flow model has been successfully implemented in the multi-physics code ALE3D. This model will provide a simulation capability for low Mach number flows where compressibility effects can be neglected. We have also completed several code enhancements to the compressible flow model as part of our investigation of applications in hemodynamics, microfluidics, and aerodynamics.

Introduction

The need for robust multidisciplinary modeling tools is increasing. ALE3D provides many of the physics options required, but not all. The area in which the deficiency is the greatest is in the treatment of low Mach number flows where a compressible flow model with explicit time integration is inefficient. The objective of this project at Lawrence Livermore National Laboratory (LLNL) is to provide an improved modeling capability for the low Mach number area by incorporating an incompressible flow module in ALE3D. With this capability, many programmatic problems of immediate and future concern can be solved. Examples include modeling of vehicle aerodynamics and underhood thermal transport for reducing energy consumption, modeling near-field biological or chemical dispersion for counterproliferation and missile defense projects, modeling of arterial hemodynamics for medical applications, microfluidics modeling, and thermal convection in storage facilities for the defense programs.

Additional improvements in ALE3D algorithms allow the code to better support modeling needs in microfluidics and medical fields. LLNL's Medical Technology Program has identified a need for detailed modeling of arterial flow as a means of understanding the mechanisms that control the formation of aneurysms. The use of arterial inserts,

or stents, to affect arterial structure or evolution is now becoming common. The interaction of these artificial structures on flow dynamics is not well understood and is currently not adequately characterized in the design process.

The term hemodynamics has been used to describe this general technical area. Such modeling is best performed with the incompressible model and will require close coupling to the dynamics of the arterial response. Models to describe the viscous flow properties of blood as well as the arterial response will have to be developed. The existing explicit, compressible algorithms in ALE3D are being used to address modeling issues and determine the applicability of the modeling approach while the incompressible capability is under development.

The microfluidics work, particularly with respect to the representation of fluid properties, has considerable synergism with modeling activities in the arterial flow arena. An initial ALE3D calculation of a piezoelectric crystal-driven fluid indicated potential utility for LLNL's Acoustically-Driven Microfluidic System Project.¹ The initial microfluidic investigations were also performed with the explicit, compressible algorithms in ALE3D.

Much of the computational fluid dynamics (CFD) parallelization and the numerical solution methods are new in this work, as well as the multi-physics coupling of the thermal/fluid models and

structural/fluid models. (The flow, diffusion, and structural phenomena have different stability and accuracy criteria which direct the numerical solution approach, grid resolution, and time step.) The ALE3D code is at the forefront of multi-physics modeling and its capabilities and demonstrated applications in massively parallel computing are unmatched by currently available commercial and non-commercial software. The addition of an incompressible flow model coupled to its structural, heat transfer, and chemistry models pushes us into previously unattainable areas of computational modeling.

Progress

Incompressible Flow Model

An Eulerian formulation of the time-dependent incompressible Navier-Stokes equations is solved using a finite-element approach. The finite-element method (FEM) used is similar to that developed by Gresho *et al.*² Unlike their original investigations our algorithms allow for the use of unstructured meshes with one-point integration, hour-glass stabilization, and balancing tensor diffusivity (BTD) for explicit time integration.

Using Galerkin FEM, the discretized continuity and momentum equations can be written in matrix form as

$$\mathbf{C}^T \mathbf{u} = 0 \quad (1)$$

$$\mathbf{M} \dot{\mathbf{u}} + (\mathbf{K} + \mathbf{N}(\mathbf{u})) \mathbf{u} + \mathbf{C} \mathbf{p} = \mathbf{F} \quad (2)$$

where \mathbf{u} is the nodal velocity vector, \mathbf{p} is the pressure vector, \mathbf{M} is the mass matrix, \mathbf{K} is the diffusivity, $\mathbf{N}(\mathbf{u})$ is the advection operator, \mathbf{C} is the gradient operator, and \mathbf{F} is the user supplied natural boundary condition.

In the current implementation, a lumped mass matrix is used and the coefficient matrices are generated using one-point Gaussian quadrature. The Q1P0 element formulation is used, which provides trilinear velocity support with piecewise constant pressure in three dimensions. The equation solution method was chosen for compatibility with the solver finite element interface³ developed by Sandia National Laboratories (SNL) in collaboration with LLNL. The continuity and momentum equations are solved together for the velocity difference and pressure with an explicit forward Euler time integration

$$\begin{bmatrix} \mathbf{M} & \mathbf{C} \\ \mathbf{C}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{v} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{0} \end{bmatrix} \quad (3)$$

where

$$\mathbf{f} = \mathbf{F} - (\mathbf{K} + \mathbf{N}(\mathbf{u})), \text{ and} \quad (4)$$

$$\mathbf{v} = \frac{\mathbf{u}^{n+1} + \mathbf{u}^n}{\Delta t} \quad (5)$$

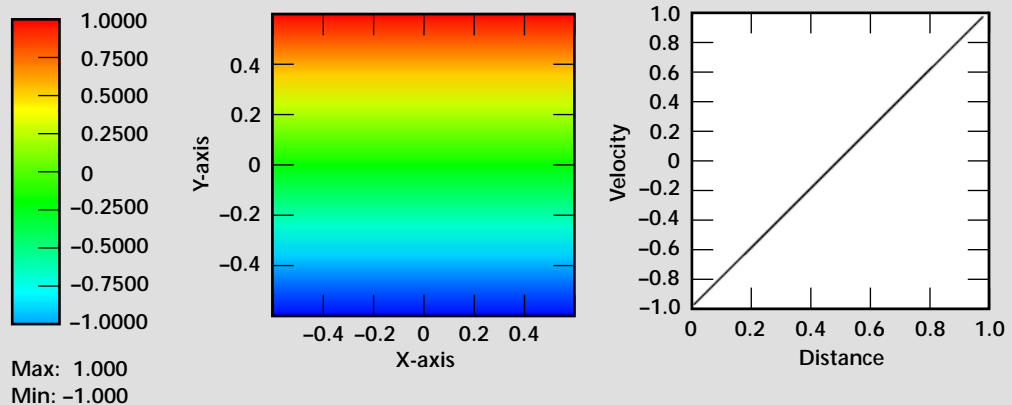
where \mathbf{n} is the time step level.

An additive correction (diffusivity) to the diffusion matrix balances the negative diffusion induced by the explicit Euler time integration (that is, BTD, Gresho *et al.*²). Also an hour-glass correction is added to the one-point quadrature diffusion matrix to damp any zero energy modes that may be present because of the reduced integration scheme (Goudreau and Hallquist⁴ and Gresho *et al.*²). To reduce the computational effort in the evaluation of the advection term, we use a “centroid advection velocity” simplification as was done by Gresho *et al.*²

For turbulent flow modeling a large-eddy simulation (LES) capability is available. In an LES of turbulent flows, the large-scale motion is calculated

Figure 1.
Incompressible flow
simulation of shear-
driven flow.

X - component of velocity



explicitly (that is, resolved) and the small-scale motion is modeled (that is, approximated with semi-empirical relations). For more details on the model see McCallen *et al.*⁵

Figures 1 and 2 are example simulations for a shear-driven flow and a lid-driven cavity, respectively. The shear driven flow was one case used for model verification.

Compressible Flow Model

Several compressible flow model enhancements were completed for application to flow around a circular cylinder, a hemodynamics application of pulsating flow in an artery with a stent, shock impact on a droplet, and two microfluidics applications of supersonic flow in a micro-nozzle and acoustically driven micro-channel flow.

Verification: Flow around a Circular Cylinder.

The simulation of laminar flow around a circular cylinder at Reynolds number, Re , of 1200 and Mach number, Ma , of 0.3 is shown in **Fig. 3**. (The Reynolds number is a measure of inertial forces to viscous forces: $Re = UD/\nu$, where U is the freestream velocity, D is the cylinder diameter and ν is the fluid kinematic viscosity.) The calculated drag coefficient, C_D , and Strouhal number, St , for our 2-D and 3-D calculations agree with those of Cox *et al.*⁶ (The drag coefficient, C_D , is a dimensionless drag force defined as the drag force/(dynamic pressure x projected area). The Strouhal number is defined as the oscillation frequency over the mean speed and is a measure of the vortex shedding frequency.) In 2-D, the calculated $C_D = 1.6$ and $St = 0.23$. In 3-D, $C_D = 1.4$ and $St = 0.20$.

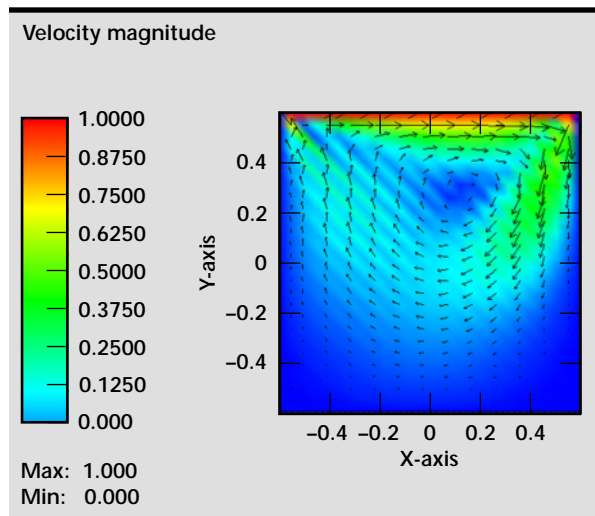


Figure 2. Incompressible flow simulation of “non-leaky” lid-driven cavity (that is, slip at top boundary with no-slip at top corners).

Hemodynamics: Pulsating Flow through an Arterial Stent. Modeling the effects of a stent in an artery requires the coupling of a pulsating blood flow and the response of the artery wall. **Figure 4** is an example calculation of pulsating flow through an arterial stent. In this first attempt, the stent is modeled with a simple material change. The region of the artery with the stent is a stiff material compared to the more flexible artery wall region.

Microfluidics: Supersonic Flow in a Micro-Nozzle and Acoustically-Driven Micro-Chamber Flow.

ALE3D’s capability as a microfluidics analysis tool was evaluated by application to two sample problems. The first case is a supersonic micro-nozzle flow that was developed and investigated by Massachusetts Institute of Technology for use by the spacecraft industry to drive low-thrust propulsion systems.⁷ **Figure 5** shows the calculated Mach number distribution for $Re = 2805$. The

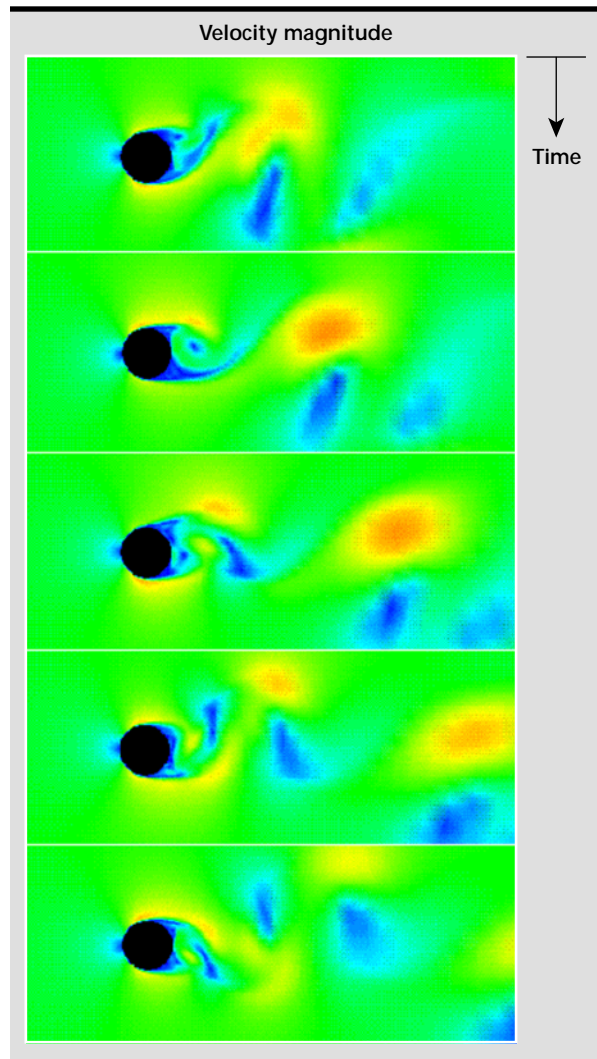
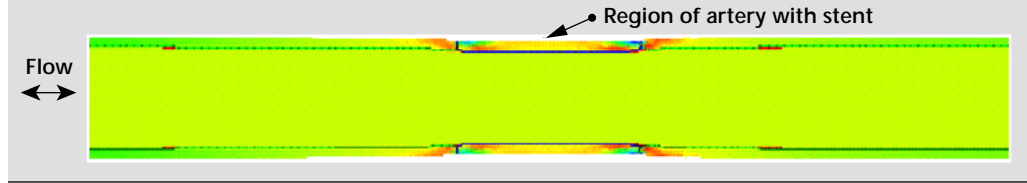


Figure 3. Simulation of turbulent vortex shedding behind a circular cylinder.

Figure 4. Simulation of an artery and stent with pulsating flow and the calculated relative stress.



challenge for this nozzle simulation is to capture the reduction in mass flow due to viscous effects at the wall (that is, flow blockage due to a substantial boundary layer or velocity gradient at the nozzle wall).

The second case is an acoustic micro-chamber. The modeled chamber is being developed by A. Wang at LLNL in the Center for Microtechnology. **Figure 6** shows the modeled geometry. The acoustic energy collects the macromolecules in the system. Important issues are the coupling of the driving mechanism to the chamber and the delivery and distribution of the acoustic energy.

Source Characterization: Shock Impact on a Droplet. Source characterization of the explosion and dispersion of biological or chemical substances is important to LLNL's counterproliferation and missile defense projects. One phenomenon of importance is the shock impact on a droplet and subsequent fluid breakup. **Figure 7** shows calculated droplet deformation due to shock impact.

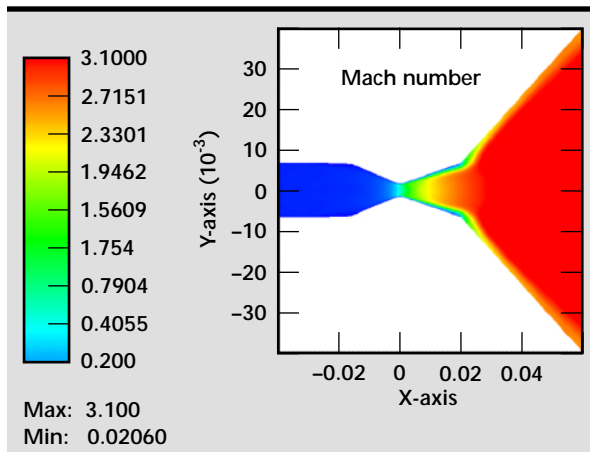


Figure 5. Simulation of flow through a supersonic micro-nozzle used for positioning of spacecraft.

Future Work

The first priority for the coming year is to complete a verification and validation study of the incompressible flow model. It is also important to investigate the code performance for parallel computing to verify that expected speed-ups are achieved.

Coupling of the incompressible flow model to the diffusion model in the ALE3D code is necessary for simulating thermal-advection problems as with applications involving the heating of an air filled room by machinery or other heat sources. For thermal-advection modeling, the addition of a buoyancy term to the flow model is required.

Capturing the expansion and contraction of an artery wall due to the pulsing blood flow will require the coupling of the incompressible flow and structural modules in the code. Plans are to use simple grid re-mapping techniques for the fluid/structure interface.

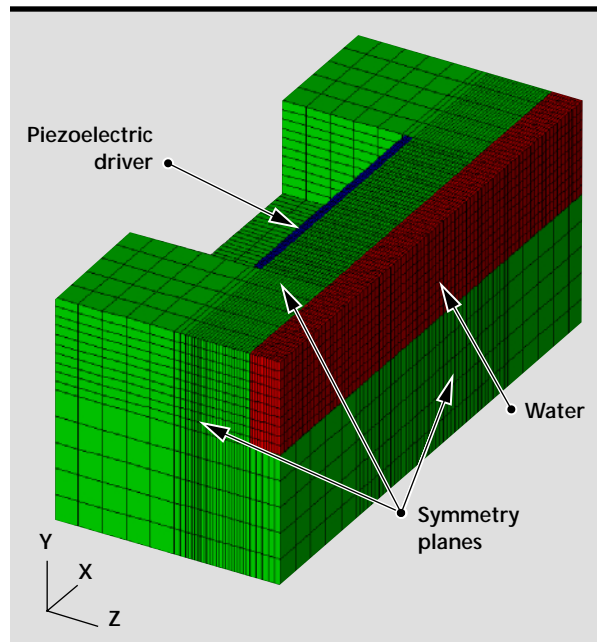


Figure 6. Model of acoustic microchamber.

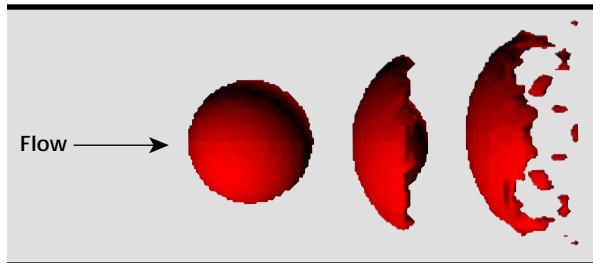


Figure 7. Simulation of shock impact on a droplet.

Acknowledgments

We gratefully acknowledge the support of C. Aro and K. Mish of LLNL and A. Williams of SNL in the implementation of the solver finite element interface. We are also grateful to P. Gresho of LLNL for his review of the project and guidance in flow modeling aspects and fluid-structure interaction modeling.

References

1. A. Wang (1999), "Acoustically-driven microfluidic system," to be published in the LLNL Annual FY99 Engineering Research, Technology, and Development Report.
2. P. M. Gresho, S. T. Chan, R. L. Lee, and C. D. Upson (1984), "A modified finite element method for solving the time-dependent, incompressible Navier-Stokes equations, Part 1: Theory," *Int. J. for Numerical Methods in Fluids*, Vol. **4**, pp. 557–598.
3. R. L. Clay, K. D. Mish, I. J. Otero, L. M. Taylor, and A. B. Williams (1999), "An annotated reference guide to the finite-element interface (FEI) specification," Version 1.0, Sandia Report SAND99-8229, January.
4. G. L. Goudreau and J. O. Hallquist (1982), "Recent developments in large-scale finite element Lagrangian hydrocode technology," *Comput. Methods in Appl. Mechanics and Eng.*, Vol. **33**, p. 725.
5. R. C. McCallen, B. T. Kornblum, and W. Kollmann (1997), "Large-eddy simulation in complex domains using the finite element method," ASME Fluids Eng. Division Summer Meeting, June 22–26, FEDSM97-3496.DOC.
6. J. S. Cox, K. S. Brentner, and C. L. Rumsey, "Computation of vortex shedding and radiated sound for a circular cylinder: subcritical to transcritical Reynolds numbers," *Theoret. Comput. Fluid Dynamics*, Vol. **12**, pp. 233–253.
7. R. L. Bayt and K. S. Breuer (1998), "Viscous effects in supersonic MEMS-fabricated micronozzles," *Proc. 3rd ASME Microfluids Symp.*, Anaheim, California, November. 